



BINDURA UNIVERSITY OF SCIENCE EDUCATION

FACULTY OF ENGINEERING

DEPARTMENT OF ELECTRONIC ENGINEERING

JUNE 2026

EEE4000 FINAL YEAR PROJECT

TITLE:

**SMART PREPAID WATER METERING SYSTEM WITH REAL-TIME MONITORING,
LEAK DETECTION AND ONLINE PAYMENT INTEGRATION**

PRESENTED BY PRECIOUS TATENDA BHASERA (B224502B)

SUPERVISOR: MS MALINDI

A project work submitted in fulfillment of the Bachelor of Science (BSc) in Electronic Engineering as a requirement of the final year project.

DECLARATION

I hereby certify that the work which is being presented in the project entitled **SMART PREPAID WATER METERING SYSTEM WITH REAL-TIME MONITORING, LEAK DETECTION AND ONLINE PAYMENT INTEGRATION** in the partial fulfillment of the requirements for the award of the degree of **BACHELOR OF SCIENCE IN ELECTRONIC ENGINEERING** and submitted in the **DEPARTMENT OF ELECTRONIC ENGINEERING, BINDURA UNIVERSITY OF SCIENCE EDUCATION**, is an authentic record of my own work carried out during the time period from **August 2022 to June 2026** under the supervision of **S Malindi, Lecturer, Electronic engineering department**.

The matter presented in this project has not been submitted by me for the award of any other degree of this or any other institute.

**Signature of the student with date
(PRECIOUS TATENDA BHASERA)**



This is to certify that the above statement made by the candidate is correct to the best of my/our knowledge.

Signature of the Supervisor of
(with date)

(S MALINDI)

A handwritten signature in blue ink, appearing to be 'S. Malindi', written on a light-colored background.

COPYRIGHT

Permission is granted to Bindura University of Science Education to produce single copies of this project and to lend such copies for private, scholarly or scientific research purposes only. The author reserves other publication rights and neither the project nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

DEDICATION

This project is dedicated to Almighty God for the strength, patience and guidance that carried me through the work.

I also dedicate it to my parents and family, whose support and sacrifices made this academic journey possible. Their encouragement helped me continue even when the work became difficult.

I further dedicate the work to my supervisor and to everyone who gave advice, encouragement or practical help during the development of the project.

ACKNOWLEDGEMENT

I thank Almighty God for the strength and wisdom that sustained me during the project.

I am deeply grateful to my parents, Mr and Mrs Bhasera, for their prayers, sacrifices and constant encouragement throughout my studies.

I also thank my supervisor, Ms Malindi, for the guidance, corrections and technical advice given during the development of this work.

Finally, I appreciate my friends, classmates and all others who assisted in small but meaningful ways during the project.

ABSTRACT

Urban councils in Zimbabwe still face practical water-management problems that are linked to manual meter reading, disputed bills, weak revenue collection, old pipe networks and slow response to leaks. Residents are affected as well because they often have no simple way of checking how much water they are using until a bill has already been issued. This project therefore focused on a prototype that brings together household metering, prepaid credit control, online payment, automatic valve operation, distribution-line leak detection, emergency credit and backup power.

The aim was to design, build and test a Smart Prepaid Water Metering System with Real-Time Monitoring, Leak Detection and Online Payment Integration for use in the Zimbabwean urban council setting. The work was treated as a prototype study, not as a finished municipal product. Its purpose was to check whether low-cost embedded hardware and a web platform could improve visibility of consumption, control unpaid usage and support quicker response to treated-water losses.

The prototype followed an engineering design and prototyping approach. The hardware used an ESP32 microcontroller connected to three YF-S201 Hall-effect flow sensors, a 12 V DC solenoid valve switched through a relay, a 16 x 2 I2C LCD, an active buzzer and a backup battery pack made from three 3.7 V lithium-ion cells protected by a 3S Battery Management System. The software side used a Flask backend, a MySQL database, HTML, CSS, JavaScript and Bootstrap dashboards, serial communication between the ESP32 and the backend, Paynow for online payment and Ngrok for remote prototype access during testing.

For leak testing, two flow sensors were installed on a simulated main distribution section and treated as points placed 500 m apart. The system compared the readings from these two points to identify a possible leak or burst in that section. A third flow sensor measured household consumption. The ESP32 used the measured consumption to deduct prepaid credit, and when the balance reached zero the solenoid valve closed. Water supply was restored only after a successful top-up or after an approved emergency-credit request updated the balance.

Testing was done under controlled laboratory conditions. The flow measurement tests gave an average accuracy of about 98%, with the largest observed error being 0.4 litres. The valve closed in less than one second after a zero-balance condition and reopened after credit was restored. A simulated leak produced a 5.10 L/min difference between the two distribution sensors and produced evidence on the dashboard, LCD,

buzzer and email alert. Ten Paynow test transactions were completed successfully, while the emergency-credit test allowed one temporary allocation and rejected a repeat request before repayment.

The results show that the proposed system is feasible at prototype level and relevant to the challenges faced by Zimbabwean councils. Its main contribution is the combination of prepaid metering, leak detection, online payment, valve control, emergency credit and battery backup in one working setup. Before any wider deployment, further work should include field testing, secure cloud hosting, stronger sensor calibration, GSM or NB-IoT communication, tamper protection, improved battery sizing and production-level data security.

ABBREVIATIONS

Abbreviation	Full Meaning
ADC	Analogue-to-Digital Converter
API	Application Programming Interface
BLE	Bluetooth Low Energy
BMS	Battery Management System
CSV	Comma-Separated Values
CNN	Convolutional Neural Network
CSS	Cascading Style Sheets
CVPA	Cost-Volume-Profit Analysis
DC	Direct Current
DT	Design Thinking
EDCDWM	Efficient Data Collection Algorithm for Water Meters
EEPROM	Electrically Erasable Programmable Read-Only Memory
ESP32	Espressif Systems 32-bit Microcontroller
GSM	Global System for Mobile Communications

GPRS	General Packet Radio Service
GPS	Global Positioning System
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
I2C	Inter-Integrated Circuit (communication protocol)
IoT	Internet of Things
IoTA4IWNet	IoT Application Framework for Intelligent Water Networks
JSON	JavaScript Object Notation
LC	Inductive-Capacitive (sensor type)
LCD	Liquid Crystal Display
LoRa	Long Range (wireless communication)
LoRaWAN	Long Range Wide Area Network
LSTM	Long Short-Term Memory (neural network)
MABE	Mobile Application Back-End
ML	Machine Learning
MQTT	Message Queuing Telemetry Transport
MySQL	My Structured Query Language (database)

NB-IoT	Narrowband Internet of Things
NPW	Negative Pressure Wave
NRW	Non-Revenue Water
PAMDES	Village Drinking Water Company (Indonesia)
PDAM	Regional Drinking Water Company (Indonesia)
PHP	Hypertext Preprocessor
PVC	Polyvinyl Chloride
PWM	Prepaid Water Meter
RFID	Radio-Frequency Identification
SMS	Short Message Service
SQL	Structured Query Language
SSL	Secure Sockets Layer
STS	Standard Transfer Specification
SWMMS	Smart Water Metering and Monitoring System
TCP/IP	Transmission Control Protocol / Internet Protocol
TelOne	Telecommunications Company of Zimbabwe
TVS	Token Vending Server

UART	Universal Asynchronous Receiver-Transmitter
URL	Uniform Resource Locator
USSD	Unstructured Supplementary Service Data
Wi-Fi	Wireless Fidelity
ZESA	Zimbabwe Electricity Supply Authority

Table of Figures

Figure 2. 1 Smart water meter using IoT	10
Figure 2. 2 Smart water meter for automatic meter reading	11
Figure 2. 3 Prepaid water meter system based on AT89S52	12
Figure 2. 4 IoT-based prepaid water system with RFID card	13
Figure 2. 5 Smart water metering system that combined an Arduino Uno microcontroller	14
Figure 2. 6 IoT smart water metering with ESP32 CAM with built-in camera	15
Figure 2. 7 Block diagram of water detection system.....	17
Figure 2. 8 System architecture of multiparameter intelligent water management system	18
Figure 2. 9 IoT-driven real-time water leakage detection system for Harare’s water distribution	19
Figure 2. 10 All 22 reviewed systems from Chapter 2	37
Figure 2. 11 Proposed System Basis.....	39
Figure 3. 1 Prototyping Methodology Illustration	42
Figure 3. 2 ESP32 Microcontroller.....	42
Figure 3. 3 Water flow sensor	43
Figure 3. 4 Solenoid Valve	43
Figure 3. 5 5v Single Channel Relay Module.....	44
Figure 3. 6 16x02 LCD with i2c	44
Figure 3. 7 12v Power Adapter	44
Figure 3. 8 Buzzer.....	45
Figure 3. 9 LM2596 Buck Converter.....	45
Figure 3. 10 Rocker Switch	45
Figure 3. 11 Veroboard.....	46
Figure 3. 12 Enclosure box	46
Figure 3. 13 Header Pins.....	46
Figure 3. 14 Header Pins.....	47
Figure 3. 15 3.7v Lithium Rechargeable Battery	47
Figure 3. 16 3s 60A Battery Management System	47
Figure 3. 17 Battery Holder	48
Figure 3. 18 Block Diagram.....	52
Figure 3. 19 Flow chart.....	60
Figure 3. 20 Interfacing esp32	62
Figure 3. 21 Interfacing flow sensors.....	63
Figure 3. 22 Interfacing solenoid valve	64
Figure 3. 23 Interfacing LCD.....	65

Figure 3. 24 Adding buzzer	66
Figure 3. 25 Power supply	67
Figure 3. 26 Firmware.....	68
Figure 3. 27 Backend API.....	69
Figure 3. 28 Database	70
Figure 3. 29 Frontend UI	71
Figure 3. 30 Overall circuit diagram.....	72
Figure 3. 31 Time to empty vs Flow rate.....	78
Figure 3. 32 Balance depletion over time	79
Figure 4. 1 Hardware prototype	83
Figure 4. 2 Full circuit diagram	84
Figure 4. 3 Web dashboard	86
Figure 4. 4 Resident logging in.....	87
Figure 4. 5 Resident dashboard.....	87
Figure 4. 6 Resident web dashboard	88
Figure 4. 7 Council dashboard	88
Figure 4. 8 Online payment request.....	89
Figure 4. 9 Paynow web page	89
Figure 4. 10 Payment processing	90
Figure 4. 11 Successful payment	90
Figure 4. 12 Declined payment	91
Figure 4. 13 Leak analytics and active meter charts	91
Figure 4. 14 Leak alerts	91
Figure 4. 15 Leak detected email alert.....	92
Figure 4. 16 LCD display.....	93
Figure 4. 17 Offline system	93
Figure 4. 18 Email leak alert.....	94
Figure 4. 19 Emergency credit request	94
Figure 4. 20 Processing request	95
Figure 4. 21 Emergency request accepted	95
Figure 4. 22 Administrator credit approval.....	96
Figure 4. 23 New credit top up updated.....	96
Figure 4. 24 Credit top up declined.....	97
Figure 4. 25 Credit top up error	97
Figure 4. 26 Daily consumption graph.....	99
Figure 4. 27 Payment history	100
Figure 4. 28 Payment history on council dashboard.....	100

Figure 4. 29 Testing emergency request	101
Figure 4. 30 Emergency credit declined	101
Figure 4. 31 Low balance warning.....	103
Figure 4. 32 Dashboard displaying zero balance	103
Figure 4. 33 Water supply suspended email alert	104
Figure 4. 34 Smart water meter back online	105
Figure 4. 35 New top up balance	105
Figure 4. 36 Active leak warning on council dashboard.....	106
Figure 4. 37 Leak alerts on council dashboard display	107
Figure 4. 38 Leak analytics	107
Figure 4. 39 Active leaks chart	108
Table 3. 1 Bill of Materials – Hardware Components	49
Table 3. 2 Circuit Components and Connections.....	72
Table 3. 3 Example calculations:	77
Table 4. 1 Sample Water Consumption Monitoring Results	97
Table 4. 2 Expanded Flow Sensor Accuracy Results	98
Table 4. 3 Credit purchase results.....	99
Table 4. 4 Emergency Credit Testing Results.....	101
Table 4. 5 Solenoid Valve Operation Results	102
Table 4. 6 Leak Detection Results	106
Table 4. 7 All four project objectives were fully met	109
Table 4. 8 System Performance Summary.....	111

Table of Contents	
DECLARATION	ii
COPYRIGHT	iii
DEDICATION	iv
ACKNOWLEDGEMENT	v
ABSTRACT	vi
ABBREVIATIONS	viii
Table of Figures	xii
Chapter 1: INTRODUCTION	1
1.0 Background of the Study	2
1.1 Problem Statement	4
1.2 Aim of the Project	4
1.3 Specific Objectives	5
1.4 Scope of the Study	5
1.5 Significance of the Study	6
1.6 Summary	6
Chapter 2: LITERATURE REVIEW	8
2.1 Introduction	8
2.2 Theoretical Background	8
2.3 Review of Related Systems and Technologies	10
2.4 Gap Analysis	37
2.5 Conceptual Framework /Proposed System Basis	38
2.6 Summary	40
Chapter 3: RESEARCH METHODOLOGY	41
3.1 Introduction	41
3.2 Research Design / Project Approach	41
3.3 System Requirements Specifications	42
3.3.1 Hardware Requirements	42
3.3.2 Software Requirements	48
3.4 System Architecture/ Block Diagram	51
3.5 System Flow Chart	59

3.6 System Design and Implementation	60
3.8 Ethical Considerations.....	79
3.9 Summary.....	81
Chapter 4: EXPERIMENTAL RESULTS AND ANALYSIS	82
4.1 Introduction.....	82
4.2 System Implementation Results.....	82
4.2.1 Hardware Prototype	83
4.2.2 Circuit Design	84
4.2.3 Resident Web Dashboard	86
4.2.4 Council Administrator Dashboard.....	88
4.2.5 Online Payment Integration.....	88
4.2.6 Leak Detection.....	91
4.2.7 Emergency Credit Facility	93
4.3 Data Presentation.....	97
4.3.1 Real-time Water Consumption Monitoring.....	97
4.3.2 Online Credit Purchase Testing.....	99
4.3.3 Emergency Credit Functionality	100
4.3.4 Automatic Water Supply Control	102
4.3.5 Leak Detection Testing	105
4.4 Analysis and Interpretation	108
4.4.1 Performance Evaluation and System Limitations	110
4.5 Discussion of Findings	112
4.6 Summary.....	113
Chapter 5: CONCLUSION AND RECOMMENDATIONS.....	114
5.1 Summary of the Study	114
5.2 Achievement of Objectives	115
5.3 Limitations.....	115
5.4 Recommendations for Future Work	116
5.5 Chapter Summary	117
References.....	118
Appendices.....	120

Chapter 1: INTRODUCTION

Access to clean and affordable water is a basic need, but many urban residents in Zimbabwe still experience unreliable supply and disputed billing. Urban councils are working with ageing infrastructure, leaking distribution pipes, inaccurate meter readings and a growing amount of unpaid water revenue. The post-paid billing approach used by many councils depends on manual readings that are not always taken regularly or accurately. This creates disputes between residents and councils, weakens trust and reduces the money available for repairs and upgrades. As urban populations increase, these problems place more pressure on infrastructure that is already difficult to maintain.

The issue is not only that better technology exists elsewhere. The problem is that most available systems do not deal with the Zimbabwean situation as a whole. Residents often dispute estimated bills and delay payment, while councils continue losing treated water through leaks and burst pipes before it reaches consumers. Akimu et al. (2025) showed that real-time IoT monitoring can support leak detection and reduce non-revenue water in Harare. Other studies, including Purboyoy et al. (2024), Ntambara et al. (2023) and Taufik et al. (2020), show that flow sensors, automated valves, online platforms and prepaid metering can improve water management. However, many of these systems focus on one part of the problem and do not combine billing, resident monitoring, emergency access and distribution leak detection in one locally relevant platform.

This project was developed to respond to that gap using a practical prototype. The proposed Smart Prepaid Water Metering System with Real-Time Monitoring, Leak Detection and Online Payment Integration measures household consumption through flow sensors and controls water supply with a solenoid valve. When prepaid credit is available, water is supplied normally. When the balance reaches zero, the valve closes. The system also includes an emergency-credit option because prepaid water control can become unfair if vulnerable users are cut off without any temporary relief mechanism. For leak detection, the system compares flow readings from two points on the distribution line and generates alerts when the difference suggests a leak or burst. Online credit purchase is included so that residents can top up without visiting council offices. A battery backup is added because power interruptions are common in Zimbabwe and the meter must remain operational during outages.

The system has two web dashboards. The resident dashboard allows users to view consumption, check balances, make payments and request emergency credit. The council dashboard allows administrators to view meter activity, monitor flow conditions, approve or reject emergency-credit requests and follow

payment or leak records. This arrangement gives both sides access to information that is normally hidden or delayed in a manual billing system.

The project is justified by the combined effect of billing disputes, water losses and weak revenue collection. When councils receive payment before consumption, the risk of unpaid bills is reduced. When flow is monitored along the distribution line, a burst caused by ageing pipes or third-party damage can be detected earlier. The system is therefore not only a billing device; it is also a monitoring and control tool. Real-time consumption information can also help residents understand their own usage and reduce waste. Online payment makes the process easier for residents, while the emergency-credit feature helps the prepaid model remain more socially responsible.

The aim of the project is to design, develop and test a prototype of a smart prepaid water metering system with real-time monitoring, leak detection and online payment integration to improve water management, revenue collection and transparency for urban councils and residents in Zimbabwe.

1.0 Background of the Study

Access to clean water remains difficult for many urban households in Zimbabwe. Councils are dealing with old pipelines, leakages, billing disputes and limited revenue for maintenance. The billing model used in many towns still depends heavily on manual meter reading and post-paid billing, which is not well suited to a growing urban population or to the need for transparent service delivery.

Manual meter reading can lead to estimated bills, delayed readings and errors. When residents do not trust the bill, they are less willing to pay. This reduces the funds available for maintenance, pumping equipment, pipe replacement and service expansion. The result is a cycle where poor billing reduces revenue, low revenue delays maintenance and delayed maintenance worsens service reliability.

A second problem is the loss of treated water before it reaches the household. Distribution pipes may burst because of age, corrosion, ground movement or accidental damage during work by other service providers such as ZESA or TelOne. Under a manual system, these bursts may only be noticed after residents complain about low pressure or after visible flooding appears. By then, the council may already have lost water that was pumped and treated at cost. Without continuous flow monitoring, repair teams also waste time trying to locate the affected section of pipe.

Electronic and embedded systems provide a practical way of addressing part of this problem. A smart meter can measure flow, process sensor signals, control a valve and send information to a software platform without waiting for manual reading. In this project, the embedded section is based on an ESP32 microcontroller, flow sensors, a relay-controlled solenoid valve and a local display. These parts are combined with web software so that the metering system can be monitored by both the resident and the council.

Previous studies support the use of smart water metering for billing and monitoring. (Taufik, et al., 2020) showed that IoT-based prepaid meters can deduct water use and control supply through a solenoid valve. (Purboyo, et al., 2024) demonstrated a prepaid meter connected to a mobile application. Mohabuth et al. (2021) and (Thanuja, et al., 2026) showed that comparing readings from two flow sensors can be used to detect leaks. These ideas are relevant to Zimbabwe, where pipe bursts and delayed leak response contribute to non-revenue water.

(Purboyo, et al., 2024) Research also shows that smart meters can improve transparency between utilities and users. When residents can see consumption data close to real time, they can better understand how credit is being used. Digital payment integration also improves convenience because users do not have to travel to a council office for every top-up. (Purboyo, et al., 2024) reported improved usability after payment gateway integration, which supports the decision to include Paynow in this project.

(Akimu, et al., 2025) Zimbabwean studies show both the opportunity and the risk of prepaid water systems. (Akimu, et al., 2025) showed that Harare can benefit from IoT-based leak monitoring and faster response to water losses. At the same time, local concerns around prepaid meters show that automatic water cut-off can harm low-income households if it is introduced without safeguards. For this reason, this project includes an emergency-credit function and treats the design as both a technical and social problem.

The proposed prototype brings these concerns together in one design. At distribution level, two flow sensors are treated as points placed 500 m apart on the main pipe, allowing a leak or burst to be identified within that monitored section. At household level, a third flow sensor measures consumption and a solenoid valve controls supply according to the prepaid balance. The software platform supports resident monitoring, council administration, Paynow top-up and emergency-credit approval. The battery backup is included so that the core control functions continue during power interruptions.

1.1 Problem Statement

Although smart water technologies have advanced globally, many Zimbabwean councils still depend on post-paid billing and manual meter reading. These systems do not give real-time consumption data, do not automatically control supply according to payment status and do not detect distribution losses before water reaches the household. As a result, councils continue facing estimated bills, resident disputes, unpaid accounts and infrastructure deterioration.

A major part of the problem is treated-water loss in the main distribution pipes. Bursts can result from old pipes, corrosion, ground movement or accidental damage during infrastructure work by organisations such as ZESA and TelOne. Since flow is not continuously monitored along the distribution line, a burst may continue for a long time before it is found. This wastes treated water and increases the financial burden on councils and residents.

Existing systems also have technical limitations. Without distribution-line flow monitoring, councils cannot easily identify where a pipe section is losing water. Without online payment, residents have fewer convenient ways to manage prepaid credit. Without automated valve control, unpaid consumption may continue. Without backup power, an electronic metering system is vulnerable to the power outages that regularly occur in Zimbabwe.

There is also an access issue. A prepaid system that simply closes the valve when credit is finished may be technically functional, but it can be socially unfair if vulnerable households have no temporary relief option. This makes emergency credit an important design requirement rather than an optional feature.

The problem addressed by this project is therefore the absence of a locally appropriate system that combines distribution leak detection, household prepaid metering, automated valve control, online credit purchase, battery backup and council-controlled emergency credit in one platform. By comparing two flow sensors placed 500 m apart, the system can identify the monitored pipe section where a leak or burst is likely to have occurred. By linking household consumption to prepaid credit and the web dashboard, the system also supports better billing control and improved transparency.

1.2 Aim of the Project

To design, develop and test a prototype of a smart prepaid water metering system with real-time monitoring, leak detection and online payment integration to improve water management, reduce

treated water losses, enhance revenue collection and promote transparency between urban councils and residents in Zimbabwe.

1.3 Specific Objectives

1. To integrate a flow sensor and a solenoid valve with ESP32 to measure water flow and automatically shut off supply when credit is depleted.
2. To design and implement a real-time leak detection system using 2 flow sensors placed apart to detect and locate pipe burst along the main distribution line by monitoring flow differences
3. To develop a flask-based web application integrated with Paynow to enable real-time monitoring, credit purchases, automatic balance updates.
4. To implement a one -time council administered emergency system enabling residents with zero balance to request top up through the web portal repayable upon next top-up.

1.4 Scope of the Study

This study covers the design, construction and testing of a prototype smart prepaid water metering system for Zimbabwean urban councils. The work is limited to two operating areas: the simulated distribution pipe and the household connection point.

At distribution level, the study covers two flow sensors treated as points placed 500 m apart on a simulated main supply line. Their readings are compared to detect a likely leak or burst in that section. The study does not cover installation across a full municipal network or the physical repair of detected leaks.

At household level, the study covers flow measurement, prepaid credit deduction, solenoid-valve control and an emergency-credit request function. It does not cover household plumbing beyond the meter point, water quality testing or pressure regulation.

On the software side, the work covers the resident dashboard, council dashboard, backend database and Paynow top-up process. It does not include a native mobile application, integration with an existing council billing system or production cloud hosting. Ngrok was used only to demonstrate remote prototype access.

All results in this report are based on controlled prototype testing. The system was not deployed in a live municipal network, and the findings should therefore be understood as prototype evidence rather than full field-performance data.

1.5 Significance of the Study

This study is significant because it addresses water billing and distribution losses using a practical embedded-systems approach. Zimbabwean councils need tools that can improve revenue collection, reduce losses and give residents clearer information without requiring expensive imported infrastructure.

For councils, prepaid metering can reduce the problem of unpaid bills because credit is purchased before consumption. The leak-detection section adds further value by helping the council identify a monitored pipe section where treated water is being lost. If a burst is detected earlier, repair teams can be sent with better information and less water is wasted while searching for the fault.

For residents, the system improves visibility. Instead of waiting for a disputed bill, a user can view balance and consumption on the dashboard. Online top-up also reduces the need to visit council offices. The emergency-credit feature is important because it prevents automated shut-off from becoming a purely punitive mechanism when a household has a temporary shortage of credit.

From an engineering point of view, the project shows how commonly available components can be combined into a working water-management prototype. The ESP32, flow sensors, solenoid valve, relay, LCD, buzzer, battery backup and web platform each perform a clear function, but their value comes from being integrated into one control and monitoring system.

The study also contributes to local smart-water research. Much of the existing literature is based on conditions outside Zimbabwe, while this project considers local issues such as power cuts, third-party pipe damage, online payment behaviour and the need for emergency water access.

1.6 Summary

This chapter explained the water-management problem addressed by the project. Zimbabwean councils lose revenue through billing disputes and lose treated water through leaks and bursts, while residents often lack real-time information about their own consumption.

The chapter also explained why a locally designed prepaid smart meter is needed. The system must consider power outages, online payment, leak detection on the distribution line and emergency access for users who temporarily have no credit.

The proposed prototype responds to these needs by combining flow sensing, solenoid-valve control, Paynow top-up, web dashboards, battery backup and emergency credit. The next chapter reviews related literature and shows how this project differs from existing systems.

Chapter 2: LITERATURE REVIEW

2.1 Introduction

Water management remains difficult for many local authorities in Zimbabwe because water losses, ageing infrastructure and estimated billing occur at the same time. Non-Revenue Water is a major concern, and electronic systems must also work in an environment where power supply is not always stable. These conditions make it necessary to review not only smart metering technologies, but also their suitability for the local operating environment.

This chapter reviews literature relevant to a prepaid water meter with web monitoring and backup power. It first discusses the theoretical background, then compares related systems, identifies gaps and finally links the literature to the proposed system basis.

- Chapter 2.2 presents the Theoretical Background, discussing the engineering theories, principles and models.
- Chapter 2.3 provides a review of related systems critically analysis existing meters.
- Chapter 2.4 presents the gap analysis, identifying the limitations in current technologies.
- Chapter 2.5 outlines the conceptual framework.

2.2 Theoretical Background

A smart prepaid water meter combines several areas of engineering. It uses sensing to measure water flow, embedded control to process signals and operate valves, communication links to transmit data and software systems to manage balances, dashboards and payments. For this project, the most relevant concepts are IoT, flow measurement, valve control, web-based monitoring and power management.

(Purboyo, et al., 2024) developed a prepaid water meter linked to the L-Connect mobile application. Their work is useful because it shows how prepaid metering can combine sensing, token handling, mobile interaction and low-power design in one device.

- Design Thinking (DT): Purboyo et al. used a user-centred design process for the L-Connect application. The stages of empathising, defining, ideating, prototyping and testing helped the researchers focus on user problems such as payment access and token redemption.

- Sensor technology: Their system used two inductive-capacitive sensors to detect water-meter disc movement. The use of two sensors at different positions helped identify movement direction and reduce reading errors caused by bounce-back effects.
- Communication protocols: The design used two communication paths rather than relying on one channel only.

1. Bluetooth Low Energy supported short-range, low-power communication between the prepaid meter and the mobile application, especially for token redemption.

- The mobile phone then used GSM connectivity to communicate with the backend for functions such as authentication and token generation. This reduced the need for separate gateway infrastructure.
- Standardisation: The system used the Standard Transfer Specification to support secure token generation. This is relevant because prepaid utility systems require a controlled and traceable method of loading credit.
- System architecture: The reviewed system used a React Native mobile application, a backend, a token vending server and a MySQL database. Communication between the front end and backend used APIs and JSON data exchange.
- Power management model: A low-power STM32 microcontroller was selected and the device operation was modelled to extend battery life. The study reported a long deep-sleep period per operating cycle so that the meter could conserve energy.
- Cost-volume-profit analysis: The same study also considered economic viability by estimating costs, revenue and the break-even point. This is important because a water meter can be technically sound but still fail if it is too expensive to deploy.

(Taufik, et al., 2020) developed an IoT-based prepaid water meter using several components relevant to this project.

- Internet of Things (IoT): The system allowed the meter to send and receive data over a network so that user quota and water usage could be monitored without manual reading.
- Embedded systems: An Arduino Nano acted as the controller. It processed sensor data, controlled the solenoid valve and communicated with the server through a Wemos module.
- Water flow sensor: The flow sensor produced a frequency signal that the controller converted into a volume measurement in litres.
- RFID: The system used an RFID card and reader to identify the customer and check the available quota from the server.

- Solenoid valve: The valve opened when quota was available and closed automatically when the quota was exhausted. This is the same basic control principle used in many prepaid water systems.
- Air vent valve: An air vent was placed before the flow sensor to reduce errors caused by air bubbles passing through the meter.
- Web development and database management: The system used a server to store user identity, quota and consumption history, while users could monitor and top up through a web platform.

2.3 Review of Related Systems and Technologies

Ahire et al. (2022) developed a prepaid water meter and water-quality checking system using IoT. The design combined a flow sensor, solenoid valve, turbidity sensor, microcontroller, web server, database and web interface. The study is useful because it connects metering with water-quality monitoring, but its dependence on stable internet access makes it less suitable for areas with unreliable power and connectivity.

Satyaraj et al. (2023) presented an IoT smart water meter for monitoring water flow and tank level. The design used a flow sensor, ultrasonic sensor and solenoid valve connected to a platform such as Blynk. The system shows that affordable sensors can support water management, but it does not provide prepaid billing, Paynow-style online payment or distribution-level leak localisation.

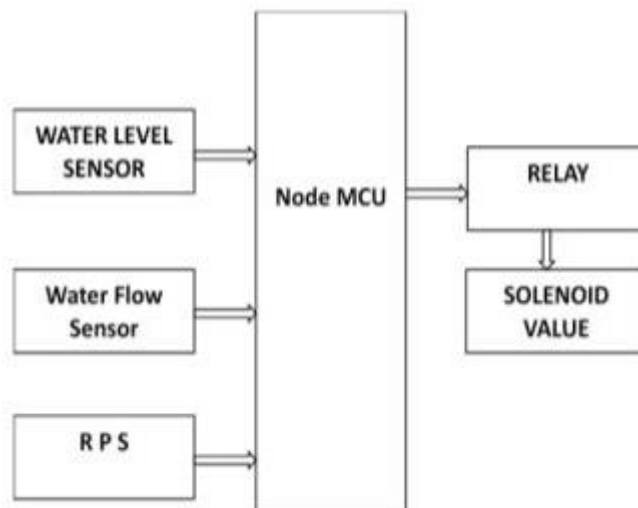


Figure 2. 1 Smart water meter using IoT

Ahmad et al. (2024) used digital image processing to automate water-meter reading for PAMDES in Indonesia. The system reduced the need for manual reading and achieved about 82% accuracy under normal conditions. Its limitation is that image quality, lighting, blurred meter faces and different meter designs can reduce accuracy.

Msisi (2022) developed an IoT water-meter reading system for Arusha City. The system used customer sensor nodes, a coordinator gateway and a cloud server to transmit readings and identify unusual consumption. It supports remote utility monitoring, but it did not include prepaid control, online payment or distribution-pipe leak localisation.

Amir et al. (2022) also examined automatic water-meter reading using an IoT-based smart meter design. In that system, a Hall-effect flow sensor measured water use and readings were sent wirelessly to a web server. The meter could also close the main valve after a set usage limit was reached. The reported measurement error was low when compared with manual measurement, which supports the use of Hall-effect flow sensors in low-cost prototypes. However, the study did not deeply address deployment cost, reliability or scalability.

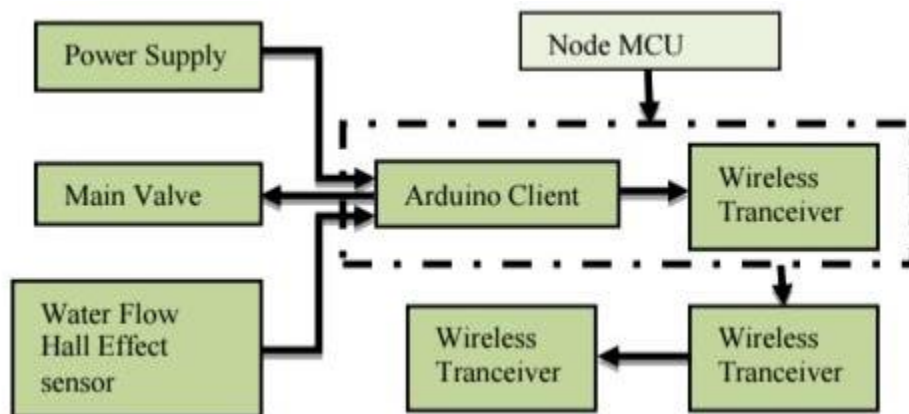


Figure 2. 2 Smart water meter for automatic meter reading

Nurhayata et al. (2021) developed a prepaid meter based on the AT89S52 microcontroller. The system stored water pulses in EEPROM and used an infrared sensor for recharge input, a flow sensor for measurement and a solenoid valve for control. It proves that a prepaid water-control loop can be implemented using simple hardware, but its offline recharge method makes it difficult to link with real-time dashboards or online payments.

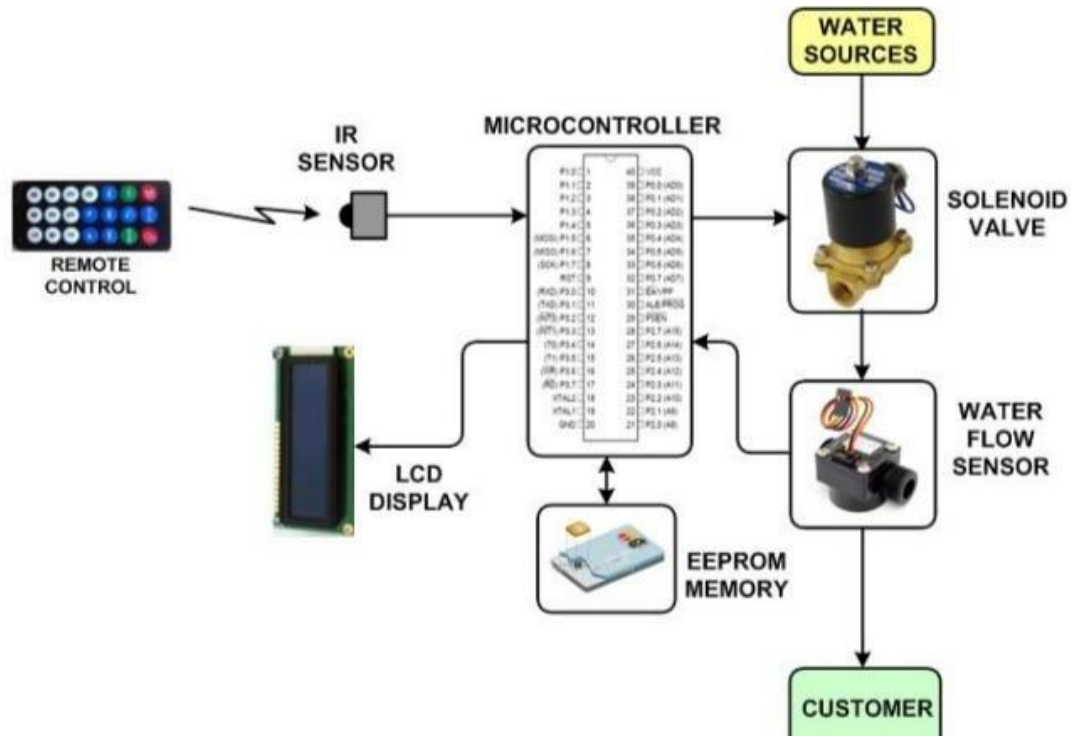


Figure 2. 3 Prepaid water meter system based on AT89S52

Purboyo et al. (2024) introduced a low-power prepaid water meter using LC sensors, BLE communication, tokenisation and a mobile application called L-Connect. The design improves payment and monitoring convenience, but data transfer to the backend is not fully real time because the meter must first connect to the mobile application before the app sends collected data to the server.

Taufik et al. (2020) proposed an IoT-based prepaid water meter that used an RFID card to identify the customer, an Arduino Nano as the controller, a Wemos board for server communication, a flow sensor for water measurement, an air vent valve to remove bubbles and a solenoid valve for supply control. Water quota was reduced according to measured usage and the valve closed when the quota was finished. The study is closely related to this project because it combines prepaid control with IoT communication.

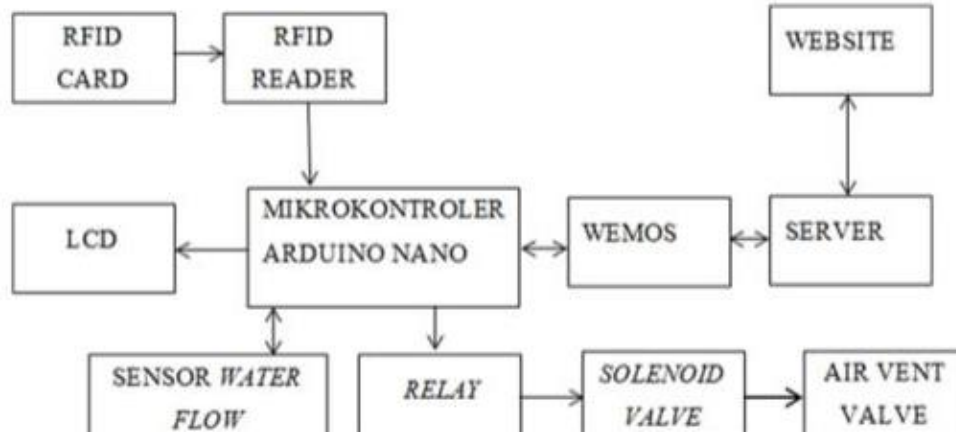


Figure 2. 4 IoT-based prepaid water system with RFID card

Ntambara et al. (2023) designed a GSM-based smart prepaid water meter with a user interface, flow sensor, automatic token generator and USSD application. Customers could load water credits using mobile money platforms and receive SMS notifications. The administrator could monitor the meter through a web interface. The work is relevant because of its prepaid and remote-control features, although it did not address distribution-line leak detection or emergency credit.

Torres-Gutierrez et al. (2026) designed a Smart Water Metering and Monitoring System for households with irregular water supply. The system used an ESP-WROOM-32E, an ultrasonic level sensor, binary flow switches, cloud services and a mobile application. Laboratory testing produced low measurement error, and household testing showed that users could monitor levels and consumption. The study is useful for real-time household monitoring, but it did not focus on prepaid valve control or council-side leak management.

Vijayasabarisiwari et al. (2025) proposed the IoTA4IWNet framework for intelligent water networks. It combined sensors, cloud computing, MQTT communication and automated control to monitor water quality and flow conditions. The framework shows the value of IoT automation in water distribution, but its reliance on stable internet and its broader water-quality focus make it different from the low-cost prepaid prototype developed in this project.

Raad et al. (2021) focused on energy-efficient smart water metering. Their system used an Arduino Uno, a YF-S201B flow sensor and an ESP8266 Wi-Fi module, while an Efficient Data Collection Algorithm reduced unnecessary data transmission. This work is important because battery life is a practical constraint in IoT metering. It also supports the need for a backup power design in the present project.

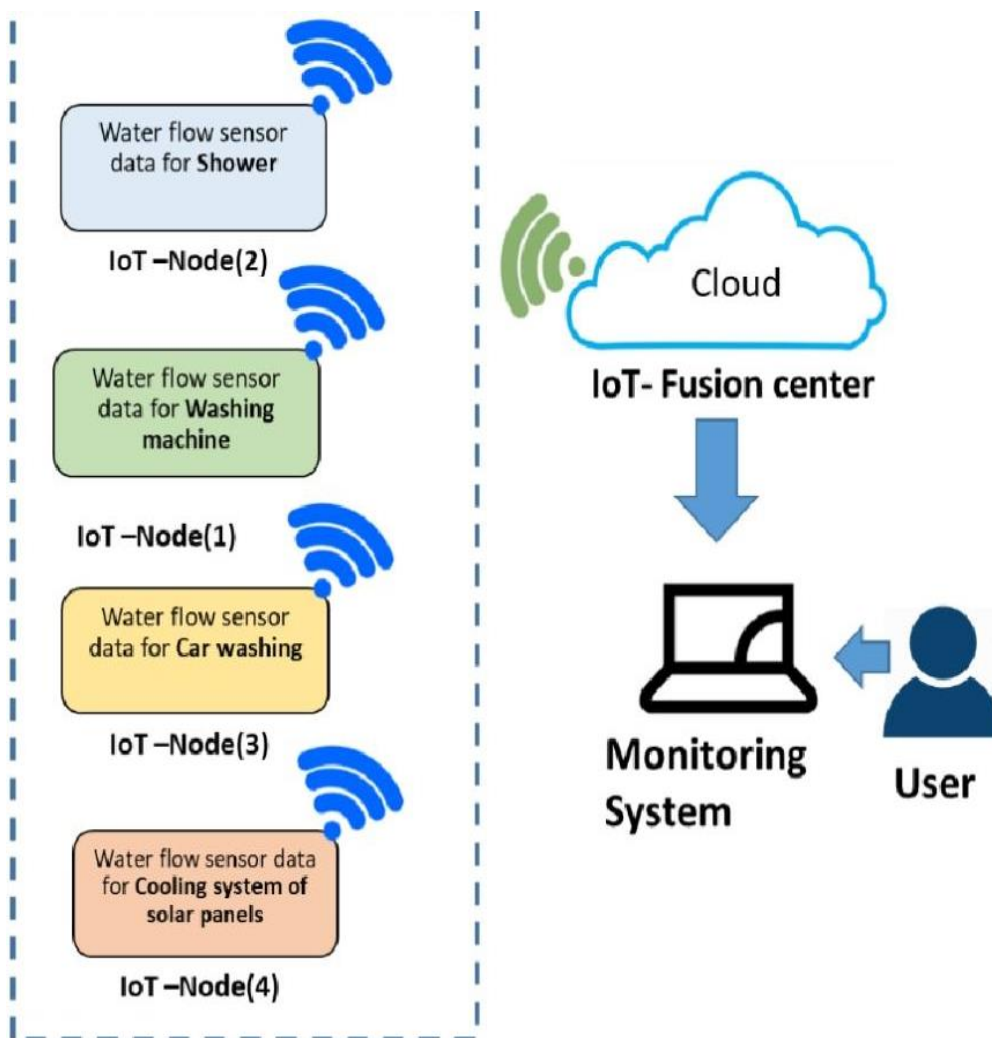


Figure 2. 5 Smart water metering system that combined an Arduino Uno microcontroller

Saw Di Wen et al. (2024) used an ESP32-CAM and a Convolutional Neural Network to read existing mechanical meter dials. This method is attractive where old meters must remain in service, but the reported recognition accuracy was not strong enough for dependable billing and automated supply control. For this project, direct flow sensing was therefore considered more suitable than image-based reading.

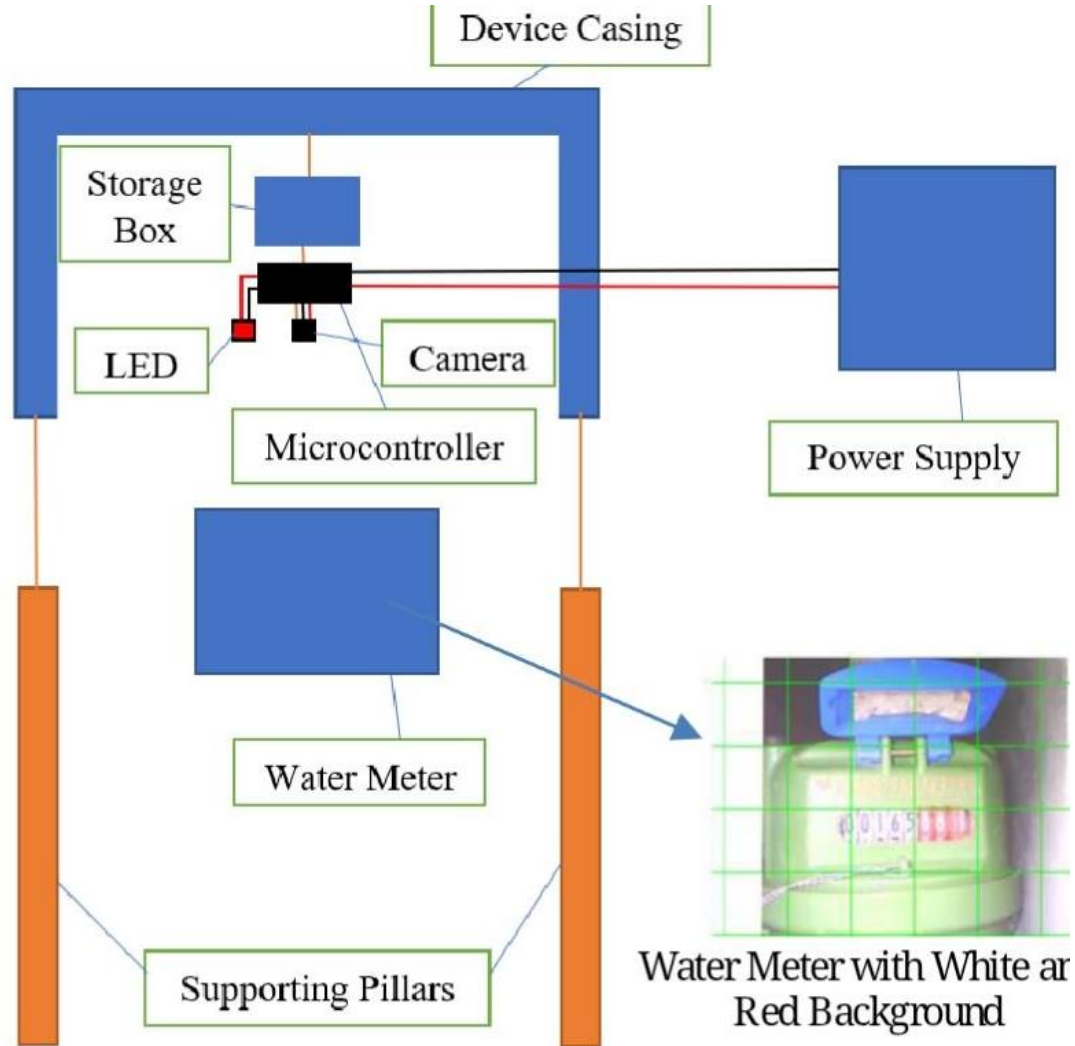


Figure 2. 6 IoT smart water metering with ESP32 CAM with built-in camera

Islam et al. (2023) used pipe vibration signals and machine learning for leak detection. A contact microphone captured sound from the pipe, the signal was converted into a spectrogram and a lightweight model classified the leak condition. The study shows that intelligent leak detection can be run on constrained

hardware, although this project uses a simpler flow-difference method because it matches the available sensors and prototype scope.

Li et al. (2024) combined negative pressure wave analysis with flow balance to improve leak localisation. Their approach is useful for larger pipeline systems because it can estimate leak position more accurately. In this project, the flow-balance principle is used as the practical starting point, while pressure-wave localisation remains a possible improvement for future versions.

Jamadarkhani et al. (2025) developed a non-intrusive smart metering framework that estimates water use from level sensors placed in tanks and reservoirs instead of installing flow meters in the pipe. The system used an ESP8266, Wi-Fi communication and cloud monitoring. Its main value is low installation cost, but the approach estimates consumption indirectly. For a prepaid control system, direct flow measurement remains more suitable because credit deduction depends on measured water use.

Soares Ascencao et al. (2023) reviewed smart water management applications and found that smart meters, system architectures and loss-management tools were major research areas. Their review supports the link between smart metering, monitoring and leak detection. It also shows that communication technologies such as LoRaWAN, SigFox and NB-IoT are important for future large-scale systems.

Nayak and Pratibha (2026) proposed an edge-intelligent IoT framework for water-use monitoring and leak detection. Instead of sending all raw readings to the cloud, the system processed recent flow data locally and used threshold logic to classify normal and abnormal flow. This is relevant because local processing can reduce delay and bandwidth use, although the present prototype uses a simpler ESP32-to-backend structure.

Thanuja et al. (2026) designed a leak-detection prototype using two flow sensors connected to an ESP32. The controller compared inlet and outlet readings and activated a buzzer, LCD warning and pump shutoff when the difference exceeded a threshold. The study supports the flow-comparison method used in this project, although the present work applies the idea to a simulated distribution section and links it with prepaid metering and council monitoring.

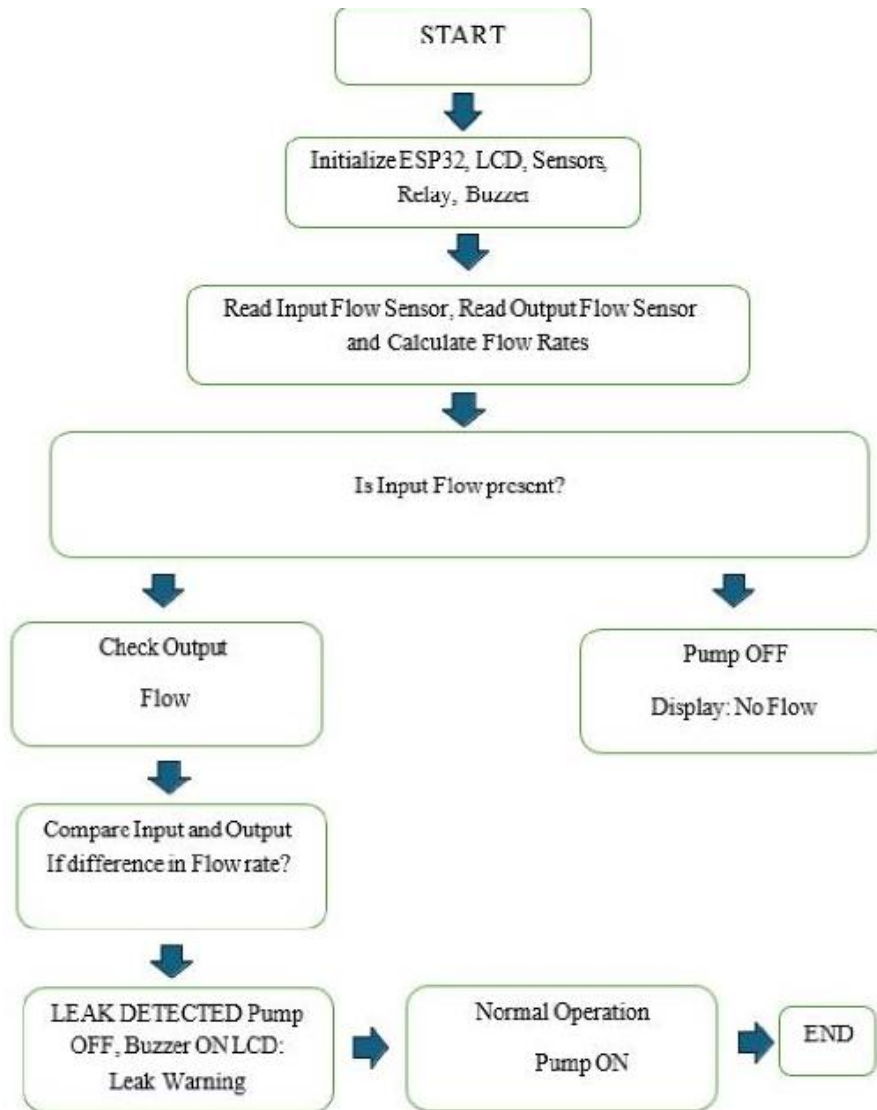


Figure 2. 7 Block diagram of water detection system

Mohabuth et al. (2021) developed a smart home leak-detection system using a NodeMCU, two flow sensors, a solenoid valve, an ultrasonic sensor and a GSM module. The system compared flow readings, sent SMS alerts and could close the valve. Its strength is the combination of detection and intervention, but it remains mainly a household-level system and does not handle council-side prepaid management.

Mendes et al. (2025) proposed a multiparameter water-management system using several sensors and a Raspberry Pi server with Node-RED and MySQL. It shows how different sensors can be brought into one monitoring platform, but the level of complexity makes it less suitable for the low-cost prepaid meter prototype required in this project.

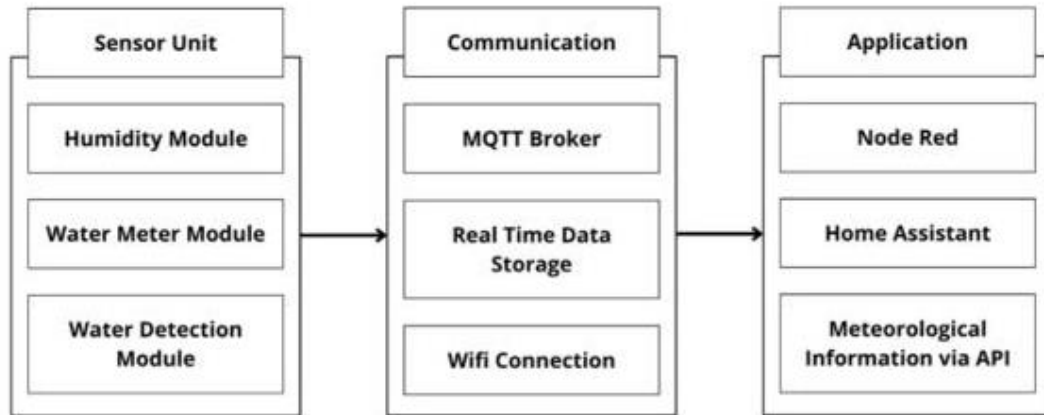


Figure 2. 8 System architecture of multiparameter intelligent water management system

Akimu et al. (2025) developed an IoT-based leakage detection system for Harare's water distribution network using ESP32 controllers, YF-S201 sensors, other supporting sensors and dashboards. This is the most locally relevant study because it deals directly with Harare and non-revenue water. Its limitation for the present research is that it focuses on leak monitoring and does not include household prepaid billing, Paynow integration, resident dashboards, automated valve control or emergency credit.

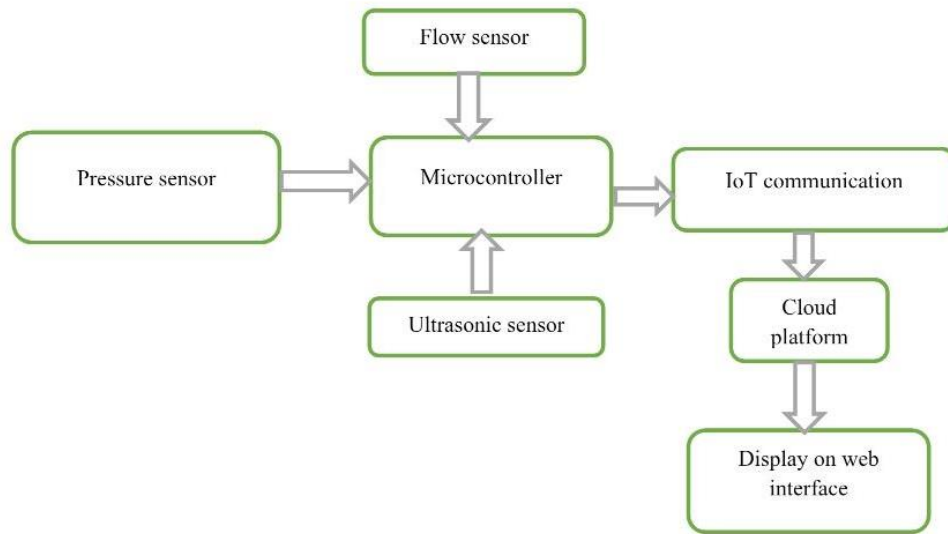


Figure 2. 9 IoT-driven real-time water leakage detection system for Harare's water distribution

CATEGOR Y A: IoT- Based Smart Water Metering Systems					
#	Author & Year	System Name / Focus	Key Hardware & Technology	Key Features & Findings	Limitations Identified
1	Ahire et al. (2022)	Prepaid Water Meter with Water Quality	• Flow sensor • Solenoid valve • Turbidity sensor	• Measures water quantity and quality simultaneously	• Fails completely if internet is unavailable • Heavy dependence

		Checking (IoT)	• Microcontroller • Web server & database • Website application module	• Turbidity sensor checks water quality • Web platform for users and water supply dept • Tracks usage, flow, and quality in real time	on stable internet connection
2	Satyaraj et al. (2023)	Smart Water Meter Using IoT (Real-Time Monitoring)	• Solenoid valve • Water flow sensor • Ultrasonic water level sensor • BLYNK IoT platform	• Real-time monitoring of water flow and tank levels • Remote monitoring via BLYNK mobile app • Prevents overflow and excessive consumption • Proactive adjustments for optimised distribution	• No advanced water quality sensors included • No integration with smart grids explored • Cannot optimise across water and energy infrastructure
3	Ahmad et al. (2024)	Digital Image Processing	• Digital camera /	• Automates meter data reading for	• Low-resolution images

		for Automated Meter Reading (PAMDES, Indonesia)	image capture device • Agile Development Method • Image processing algorithms	PAMDES village water company • Achieves up to 82% reading accuracy under normal conditions • Reduces manual meter reading workload	degrade accuracy • Blurry or non-standard meter faces cause suboptimal results • Only 82% accuracy — not reliable enough for all deployments
4	Msisi (2022)	IoT-Based Water Meter Reading System (Arusha City, Tanzania)	• Sensor nodes at customer locations • Coordinator gateway • Cloud server	• Accurate water usage measurement • Data transmitted to cloud • Flags unusual readings for authority review • Weekly usage updates sent to customers to catch billing errors early	• No direct node-to-server communication • No prepaid meter development • Limited by time and budget — many features missing

5	Amir et al. (2022)	Smart Water Meter for Automatic Meter Reading (IoT-Based)	• Hall-effect flow sensor • Solenoid valve • Web server • Smartphone access	• Measures water flow rate and total consumption • Data sent wirelessly to web server • Customer accesses data via smartphone • Automatically closes valve when user-set limit is reached • Max measurement error of only 0.03 litres	• No details on reliability or scalability provided • Cost-effectiveness not discussed • No prepaid billing or payment integration
6	Raad et al. (2021)	Energy-Efficient Smart Water Metering System with EDCDWM Algorithm	• Arduino Uno microcontroller • YF-S201B Hall-effect flow sensor • ESP8266 Wi-Fi module • Thingier.IO cloud platform	• EDCDWM algorithm reduces data transmission significantly • EDCDWM-RD approach saves ~93% energy • EDCDWM-AS approach saves ~60% energy	• No automatic valve control for leak prevention • No automated bill payment integration • Requires stable

				• Up to 99% measurement accuracy (calibration factor 6.8) • Tested across 4 real scenarios: shower, washing machine, car washing, solar cooling	internet for cloud data upload
7	Saw Di Wen et al. (2024)	IoT Smart Water Metering Using ESP32-CAM and CNN for Dial Reading	• ESP32-CAM microcontroller with built-in camera • Convolutional Neural Network (CNN) • MQTT protocol • Firebase real-time database • Node-Red gateway • Mobile application	• Reads existing mechanical meter dials automatically every 30 min • Works with existing meters — no infrastructure replacement needed • Mobile app shows consumption history and billing info • Modernises legacy meters cost-effectively	• Only 3.33% accuracy for all 8 digits on dial • 53.33% accuracy when red digits excluded — still too low for deployment • No water leak detection feature

					• No online payment system
8	Torres-Gutierrez et al. (2026)	Smart Water Metering and Monitoring System (SWMMS) for Irregular Supply Households	• ESP-WROOM-32E microcontroller • Ultrasonic level sensor • Binary flow switches • MQTT over Wi-Fi • AWS and Google Cloud Platform • Flutter mobile application	• Measurement error only 0% to 1.42% in lab testing • Real-time water level and usage monitoring • Mobile alerts for low levels, tank overflow, unusually long usage • Tested in real household over 115 days • Cloud-based data storage and processing	• Cannot separate simultaneous water usage events • Long-term battery performance not tested in real conditions • Requires stable internet and cloud services
9	Vijayasabari et al. (2025)	IoTA4IWN et Framework : Smart IoT Water Network for Real-Time Monitoring	• Smart IoT sensors (pH, turbidity, dissolved oxygen, temperature) • Flow sensors	• Monitors water quality parameters continuously • Detects unusual consumption and possible leakages	• Heavily dependent on stable internet connectivity • Does not monitor heavy metals or

			• Solenoid valves • MQTT over TCP/IP • Central cloud database	• Solenoid valves auto-control flow to reduce wastage • Reduces manual monitoring significantly • Improves billing accuracy	microbial pollutants • Not designed for areas with poor network infrastructure
10	Jamadarkhani et al. (2025)	Non-Intrusive Smart Water Metering Using Tank-Level Sensors	• ESP8266 microcontroller • Low-cost level sensors in overhead tanks/reservoirs • Wi-Fi connectivity • Cloud platform	• No flow meters on pipelines needed — uses tank level sensors instead • Predefined flow rate method for stable environments • Dynamic estimation method for variable demand • Entire system costs only ~\$75 vs ~\$750 for	• Estimation-based — less precise than direct flow measurement • Dynamic method may struggle with very rapid demand changes • Relies on Wi-Fi for cloud connectivity

				conventional meters • Effective in both lab and real-world testing	
CATEGORY B: <i>Prepaid Water Meter Systems</i>					
#	Author & Year	System Name / Focus	Key Hardware & Technology	Key Features & Findings	Limitations Identified
11	Nurhayata et al. (2021)	Prepaid Water Meter System Based on AT89S52 Microcontroller	• AT89S52 microcontroller • EEPROM memory (stores water pulse data) • Infrared sensor (remote control recharge) • Flow sensor • Solenoid valve	• Pulse-based water credit system • EEPROM stores balance even without power • Remote control used to recharge water pulses • LCD displays current water usage and balance	• Fully offline — no internet connectivity • If remote control is lost or breaks, users cannot recharge • No remote monitoring, usage analytics, or

			• LCD display	• Solenoid valve closes automatically when balance depletes	utility integration • No online payment capability
12	Purboyo et al. (2024)	Low-Power Prepaid Water Meter with LC Sensors and BLE Mobile App (L-Connect)	• LC (Inductive-Capacitive) sensors • Bluetooth Low Energy (BLE) • STM32L431C BT6 low-power microcontroller • React Native mobile app (L-Connect) • Token Vending Server (C language) • Laravel backend + MySQL • STS tokenisation (IEC 62055)	• Innovative low-power design targeting 6-year battery life • Tokenisation via Standard Transfer Specification (STS) protocol • BLE connectivity between meter and mobile app • L-Connect app for token redemption and monitoring • Mathematical power model used to optimise sleep/wake cycles • Break-even at 84 units	• Data to back-end server is NOT real-time • BLE connection must be established before data can be uploaded • No cloud or Wi-Fi direct upload from meter

				(CVPA analysis)	
13	Taufik et al. (2020)	IoT-Based Prepaid Water Meter with RFID Card	• Arduino Nano microcontroller • Wemos Wi-Fi board • RFID card and reader • Solenoid valve • Air vent valve • Water flow sensor	• RFID card identifies customer and verifies quota on server • Air vent valve removes air bubbles for accurate flow measurement • Solenoid valve opens when quota is sufficient and closes when exhausted • Flow sensor calibrated against a conventional water meter • Server stores customer ID, quota, and consumption history	• Manual process required to recharge water quota • No automated top-up or online payment integration • Depends on server and internet connection for RFID verification
14	Ntambara et al. (2023)	GSM-Based Smart Prepaid	• Water flow sensor	• Customers register and top up via Airtel Money,	• No leak detection or prediction capability

		Water Meter with USSD Mobile Money (Tanzania)	• Solenoid valve • GSM module • LCD display • USSD application • PHP, HTML, JavaScript, SQL web interface	TigoPesa or Mpesa • USSD app allows credit loading without smartphone • SMS notifications when tokens are about to expire or successfully loaded • Administrator manages system remotely via web interface • Valve closes automatically when credits are exhausted	• No resident-facing web dashboard — only admin web interface • Future research needed for AI/ML-based leak prediction and cybersecurity
CATEGORY C: Water Leakage Detection Systems					
#	Author & Year	System Name / Focus	Key Hardware & Technology	Key Features & Findings	Limitations Identified

15	Islam et al. (2023)	IoT + Machine Learning Based Water Leakage Detection (On-Device, Cloudless)	• Piezoelectric contact microphone • ESP32 microcontroller • Frequency Domain Artificial Neural Network (11 KB model) • ESP-NOW low-power communication protocol • 3500 mAh battery	• Detects leaks from pipe vibration sounds converted to spectrograms • Model quantized and pruned to 11 KB — runs on ESP32 with only 4 KB RAM • No cloud dependency — processes entirely on-device • 100% leak detection accuracy across 10 real-world test scenarios • Only 5.46 mA sleep current — operates 25+ days on a single battery	• Vibration-based method differs from flow differential approach • Not suitable for distribution-level leak localisation • Limited scalability to large networks without additional infrastructure
16	Li et al. (2024)	Combined NPW and Flow Balance Method for	• Pressure sensors (at strategic	• Combines flow balance (mass conservation) and NPW	• Simulation-based study — real-world

		Pipeline Leak Detection and Localisation	pipeline points) • ANSYS Fluent simulation • Negative Pressure Wave (NPW) analysis • Flow balance / mass conservation model	methods in one system • Flow balance reduces false alarms from pump switching or pressure changes • Average leak positioning error of only 4.76% • Four evenly spaced sensors gave best localisation accuracy in simulation • Sensors closer to leak give more accurate results	deployment not fully tested • NPW signal weakens with distance from leak reducing remote accuracy • No prepaid billing or consumer-facing features
17	Thanuja et al. (2026)	Smart Water Leakage Detection and Monitoring Prototype (Dual Flow Sensor, ESP32)	• ESP32 microcontroller • Two flow sensors (inlet and outlet) • Relay module (pump control)	• Dual-sensor flow balance: continuously compares inlet vs outlet flow • Activates buzzer alarm and LCD	• No prepaid billing or payment integration • No distribution-network

			• Buzzer alarm • LCD display • Blynk mobile application	warning when leak detected • Automatically switches off water pump via relay • Real-time alerts sent via Blynk mobile application • Reliable under both normal and intentional leak conditions • Low-power design suitable for battery/solar deployment	scale deployment • Focused only on household pipeline level
18	Mohabuth et al. (2021)	IoT-Based Smart Home Water Leak Detection System	• NodeMCU microcontroller • Two water flow sensors • Solenoid valve • Ultrasonic sensor	• Dual flow sensor comparison for leak detection • Ultrasonic sensor catches small dripping leaks missed by flow sensors • SMS alert sent via GSM	• Single microcontroller limits number of sensors • Not scalable to large deployments without

			• GSM module • ThingSpeak cloud platform • Android application	when leak detected • Android app shows real-time sensor readings via ThingSpeak • Auto-closes solenoid valve if user does not respond within 5 minutes • Follow-up confirmation message sent after auto-valve closure	hardware redesign • No prepaid billing or online payment
19	Nayak & Pratibha (2026)	Edge-Intelligent IoT Framework for Smart Water Monitoring and Leak Detection	• Edge computing layer (on-device processing) • IoT flow sensors • IoT cloud platform (for visualisation/alerts) • Sliding time-window algorithm	• Processes data at the edge — not raw-uploaded to cloud • Sliding time-window reduces false alarms from temporary flow fluctuations • Three-tier classification: normal (<5.0 L/min), warning (5.0-	• Simulation-based results — real-world physical testing not reported • No prepaid billing, payment, or valve control integration

			• Threshold-based decision classifier	6.5), confirmed leak (>6.5) • Reduces network bandwidth usage significantly • Modular design supports homes, commercial buildings and smart cities	• Threshold values may need tuning for different environments
CATEGORY D: <i>Multiparameter & Systematic Review Studies</i>					
#	Author & Year	System Name / Focus	Key Hardware & Technology	Key Features & Findings	Limitations Identified
20	Mendes et al. (2025)	Multiparameter Intelligent Water Management System (IoT + ML + Weather API)	• ESP32-CAM module (meter reading) • Arduino MKRWiFi 1010 boards • Soil moisture,	• Camera-based automated meter reading • Soil moisture and humidity sensors detect outdoor leaks	• High complexity — many components to integrate and maintain • Camera-based

			humidity, and water leak sensors • MQTT protocol • Raspberry Pi + Node-Red server • MySQL database • OpenWeather API	• OpenWeather API distinguishes rainfall from actual water leaks • Machine learning identifies abnormal consumption patterns • Real-time monitoring, remote alerts, and automated data analysis	reading accuracy not fully quantified • No prepaid billing or payment gateway integration
21	Akimu et al. (2025)	IoT-Driven Real-Time Water Leakage Detection for Harare's Distribution Network	• ESP32 microcontrollers • YF-S201 Hall-effect flow sensors • Ultrasonic sensors • Pressure sensors • Cloud-based monitoring platform	• Three-layer architecture: sensing, network, application • Compares acoustic, vibratory, drone imaging and LoRa IoT detection methods • NB-IoT identified as most suitable for long-range, energy-	• No prepaid billing system included • No online payment integration • No automatic valve shutoff • No consumer-facing resident

			• Web-based dashboards • MQTT, HTTP, Wi-Fi, LoRaWAN, NB-IoT reviewed	efficient deployment • Real-time dashboards enable faster leak response times • Directly relevant to Zimbabwe's distribution network challenges	web application • Focused solely on council-side leak monitoring
22	Soares Ascencao et al. (2023)	Systematic Literature Review: Applications of Smart Water Management Systems (62 Articles)	• Systematic review methodology • LoRaWAN, SigFox, NB-IoT communication technologies reviewed • Smart meter data analysis frameworks	• Reviewed 62 research articles and identified 100 application examples in 10 categories • Smart water meters most common application (43% of articles) • Implementation models/architectures at 15%; loss	• Review paper only — no new system prototype developed • Most reviewed systems focus on distribution; treatment and reuse under-researched • Communication technology comparisons may not

				management at 11% • NB-IoT identified as most scalable communication technology • Strong link between smart metering and water loss management confirmed • Gaps found in water treatment and wastewater reuse research	reflect rapidly evolving standards
--	--	--	--	--	------------------------------------

Figure 2. 10 All 22 reviewed systems from Chapter 2

2.4 Gap Analysis

The first major gap is that the reviewed systems do not combine distribution-line leak detection with household prepaid metering in one platform. Mohabuth et al. (2021) and Thanuja et al. (2026) show that comparing two flow sensors can detect leaks, but their work is mainly at household or small-pipe level. Akimu et al. (2025) focuses on Harare's distribution network, but it does not include prepaid billing, online top-up, valve control or a resident interface. This project links both sides by monitoring a simulated 500 m distribution section while also controlling household prepaid supply.

The second gap is power and connectivity resilience. Several reviewed systems depend heavily on stable internet access for monitoring and alerts. In Zimbabwe, power cuts and unstable connectivity can affect an

electronic meter. The proposed design addresses this by using battery backup for the core hardware and by keeping sensing, valve control and local processing available during power interruptions.

(Nurhayata, et al., 2021)(Ntambara, et al., 2023)(Purboyo, et al., 2024)The third gap relates to payment. Some reviewed systems use offline token or remote-control recharge, while others use mobile money or mobile applications. In this project, Paynow is integrated with the resident dashboard so that a successful payment updates the database and allows the hardware to restore supply through the control logic. This is important because payment confirmation must be linked directly to the user's water balance. The fourth gap is social protection. Water is an essential service, so prepaid control needs safeguards. Literature on prepaid systems in Zimbabwe raises concern about automatic cut-off for vulnerable users. The proposed system therefore includes a one-time emergency-credit request that is reviewed by a council administrator and recovered on the next top-up.

(Torres-Gutierrez, et al., 2026) (Taufik, et al., 2020) (Ntambara, et al., 2023) (Akimu, et al., 2025)The fifth gap is the separation between resident and council needs. Some systems focus on the consumer interface, while others focus only on utility monitoring. This project provides both a resident dashboard and a council dashboard, allowing residents to view usage and payments while council staff monitor meters, leaks and emergency-credit requests.

(Akimu, et al., 2025) The final gap is local fit. Many reviewed systems were designed for environments that do not have the same combination of power interruptions, third-party pipe damage, council revenue pressure and social concerns around water access. The proposed system is therefore designed around Zimbabwean urban water conditions rather than treated as a generic smart-metering product.

2.5 Conceptual Framework /Proposed System Basis

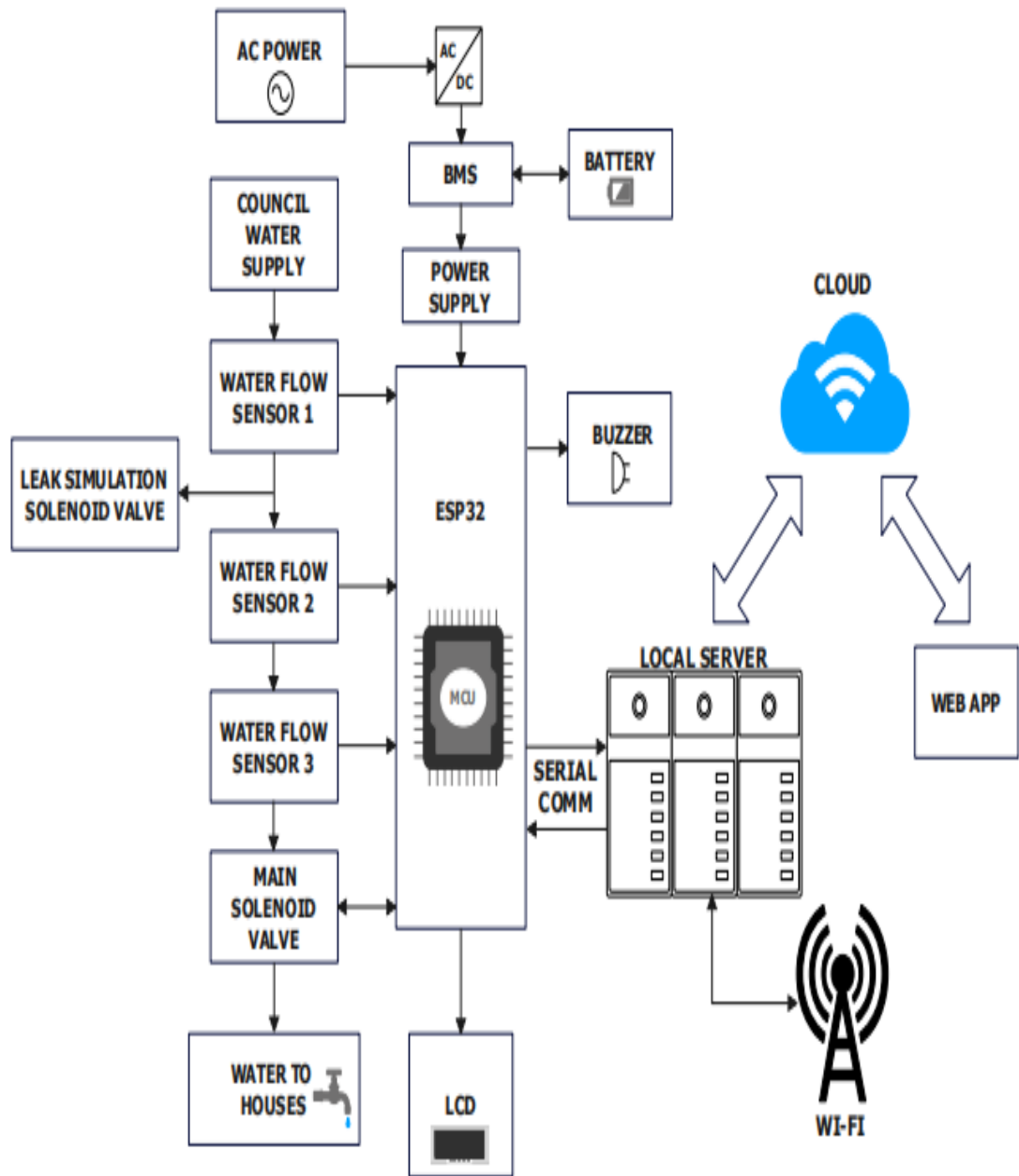


Figure 2. 11 Proposed System Basis

2.6 Summary

The reviewed literature confirms that smart water metering can improve consumption monitoring, billing accuracy and leak detection. Studies by Purboyo et al. (2024), Taufik et al. (2020), Thanuja et al. (2026), Mohabuth et al. (2021) and Akimu et al. (2025) each contribute useful ideas, including prepaid credit control, mobile or web interaction, flow sensing and leak detection.

At the same time, the review shows that no single system fully addresses the problem targeted in this project. Existing work either focuses on household metering, distribution leakage, online monitoring, energy efficiency or payment, but not all of these together. The proposed prototype responds to this by combining distribution-level flow comparison, household prepaid control, Paynow integration, battery backup, emergency credit and separate dashboards for residents and council administrators.

The literature therefore provides the technical basis for the prototype and also explains why a locally focused design is needed. Chapter 3 uses these findings to guide the research methodology and system implementation.

Chapter 3: RESEARCH METHODOLOGY

3.1 Introduction

This chapter explains how the prepaid water meter prototype was designed, built and tested. It covers the hardware, software and testing approach used to develop the web-based prepaid meter with leak detection and backup power.

The chapter also explains the system requirements, block diagram, control logic and implementation stages. It follows the movement of data from the flow sensors to the ESP32, then to the Flask backend, database and user dashboards.

3.2 Research Design / Project Approach

The project used an engineering design and prototyping approach because the work required both physical hardware assembly and software development.

The process started by identifying the weaknesses of manual and post-paid water billing, including inaccurate readings, delayed payments and limited real-time monitoring. From these problems, the main requirements were defined: prepaid control, flow measurement, leak detection, web monitoring and backup power.

The design stage involved choosing the ESP32, flow sensors, solenoid valve, relay, LCD, battery backup and web software stack. The prototype was then assembled and tested in stages so that hardware and software problems could be corrected as they appeared.

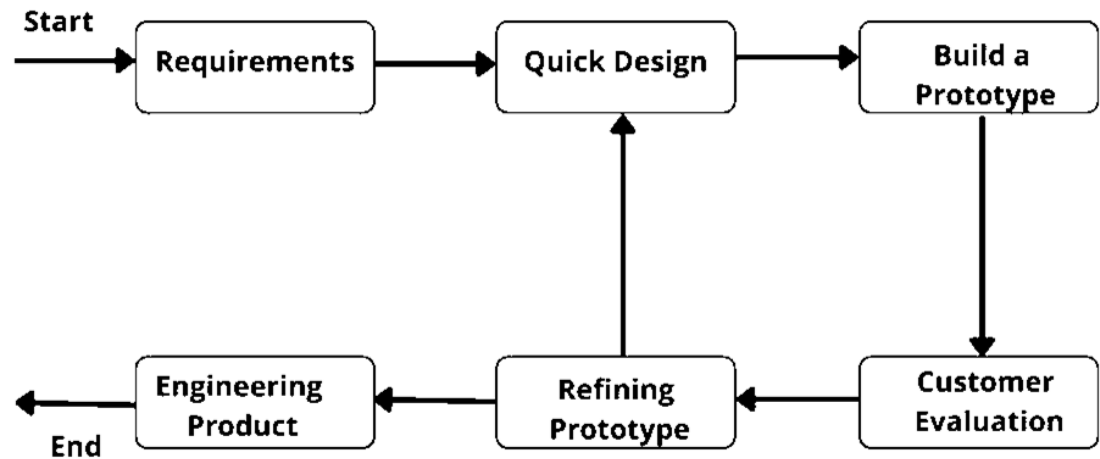


Figure 3. 1 Prototyping Methodology Illustration

After assembly, the system was tested to confirm that the main functions worked together under controlled prototype conditions.

Simulation and debugging tools were used during development to check both hardware and software behaviour. Arduino IDE was used for the ESP32 firmware, while the backend and dashboards were tested through the local server setup before full prototype testing.

3.3 System Requirements Specifications

3.3.1 Hardware Requirements

- i. **ESP32 microcontroller** – Acts as the main controller, processing sensor data and managing system operations. Its built-in Wi-Fi makes it ideal for real-time monitoring and web integration.



Figure 3. 2 ESP32 Microcontroller

- ii. **3× water flow sensors** – Measure inlet and outlet water flow to track consumption accurately. Using two sensors enables leak detection through flow comparison.



Figure 3. 3 Water flow sensor

- iii. **12V DC solenoid valve** – Controls water supply by opening or closing based on system commands. Suitable for automated prepaid water control.



Figure 3. 4 Solenoid Valve

- iv. **5V relay with optocoupler** – Allows the ESP32 to safely control the high-power solenoid valve. The optocoupler provides electrical isolation and protection.



Figure 3. 5 5v Single Channel Relay Module

- v. **16×2 LCD (I2C)** – Displays key information such as balance and usage in real-time. I2C reduces wiring complexity and pin usage.

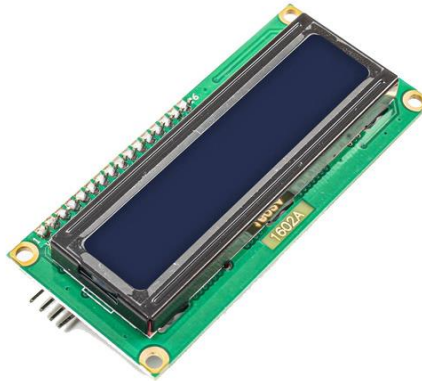


Figure 3. 6 16x02 LCD with i2c

- vi. **12V DC power adapter** – Supplies stable power to the system and high-power components. Reliable for continuous operation.



Figure 3. 7 12v Power Adapter

- vii. **Buzzer** – Provides audible alerts for events like low balance or leaks. Enhances user awareness and system responsiveness.



Figure 3. 8 Buzzer

- viii. **LM2596 buck converter** – Steps down 12V to required lower voltages (5V/3.3V). Ensures efficient and safe power distribution.



Figure 3. 9 LM2596 Buck Converter

- ix. **Rocker switch** – Enables manual control of system power. Simple and user-friendly for turning the system on/off.



Figure 3. 10 Rocker Switch

- x. **Veroboard** – Used to mount and solder components into a stable circuit. Ideal for durable prototyping.

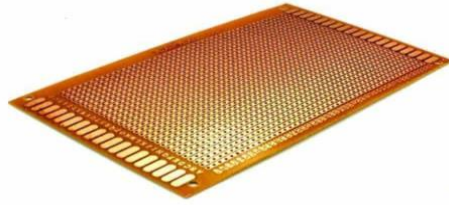


Figure 3. 11 Veroboard

- xi. **Enclosure box** – Houses and protects all components from damage and environmental factors. Improves safety and organization.



Figure 3. 12 Enclosure box

- xii. **Jumper wires** – Used for electrical connections during prototyping. Flexible and easy to modify.

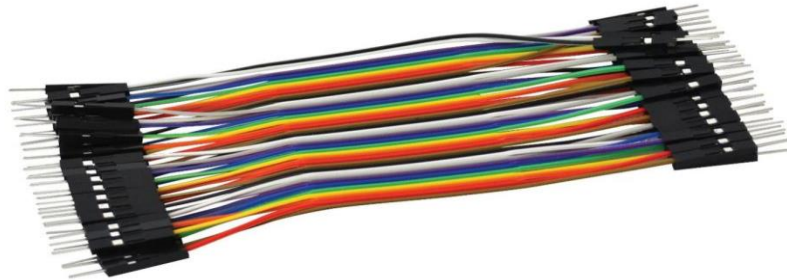


Figure 3. 13 Header Pins

- xiii. **Header pins** – Provide detachable connections between components. Useful for maintenance and upgrades.



Figure 3. 14 Header Pins

- xiv. **3.7V batteries (×3)** – Provide backup power during outages. Ensure continuous system operation.



Figure 3. 15 3.7v Lithium Rechargeable Battery

- xv. **3S 60A BMS** – Manages and protects the battery pack from overcharge, over-discharge, and short circuits. Improves safety and battery life.

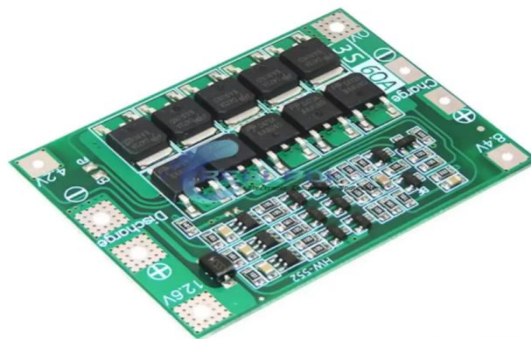


Figure 3. 16 3s 60A Battery Management System

- xvi. **3S battery holder** – Holds the batteries securely in place. Ensures proper electrical contact and organization.



Figure 3. 17 Battery Holder

Mechanical components for the structure

- Bucket (reservoir)
- Outlet fitting for tank
- Pipes (PVC conduit)
- Pipe fittings
- Manual faucet/ tap
- Cable tyres, insulation tape and thread tape
- Solvent cement

3.3.2 Software Requirements

- Arduino IDE
- Python
- PyCharm
- XAMPP
- Ngrok
- Paynow API
- Visual studio code

Table 3. 1 Bill of Materials – Hardware Components

Component	Qty	Key Specifications	Purpose in System	Role
ESP32 Microcontroller	1	Dual-core, 240 MHz, built-in Wi-Fi & Bluetooth	Central processing & control unit	Processing
YF-S201 Flow Sensor	3	Hall-effect, 1–30 L/min, 5V DC	Water flow measurement (3 points)	Sensing
12V DC Solenoid Valve	1	12V, normally closed	Automated water supply control	Actuation
5V Relay (Optocoupler)	1	5V coil, 10A contact, isolated	Switching high-power valve safely	Control
16×2 LCD (I2C)	1	I2C address 0x27/0x20, 5V	Display balance, flow, status	Output
LM2596 Buck Converter	2	Input 4–40V, output 1.5–35V, 3A	Step down 12V to 5V / 3.3V	Power

12V DC Power Adapter	1	12V, 2A minimum	Mains power supply	Power
3.7V Li-ion Battery	3	18650 cell, 3500 mAh each	Battery backup during outages	Power
3S 60A BMS	1	11.1V nominal, over-current & over-charge protection	Battery protection & management	Power
Buzzer (Active 5V)	1	5V, 30mA, 85dB	Audible alerts (low balance, leak)	Output
3S Battery Holder	1	Series 3-cell holder	Secure battery mounting	Mechanical
Veroboard	1	Stripboard for soldering	Component mounting	Mechanical
Enclosure Box	1	Waterproof or IP-rated plastic box	Component protection	Mechanical
Rocker Switch	1	SPST, 250V AC rated	Manual power on/off	Control

Jumper Wires	Set	Male-male, male-female	Prototype connections	Connection
Header Pins	Set	2.54mm pitch	Detachable component connections	Connection

3.4 System Architecture/ Block Diagram

The architecture is derived from the conceptual framework presented in Chapter 2.

Overall block diagram

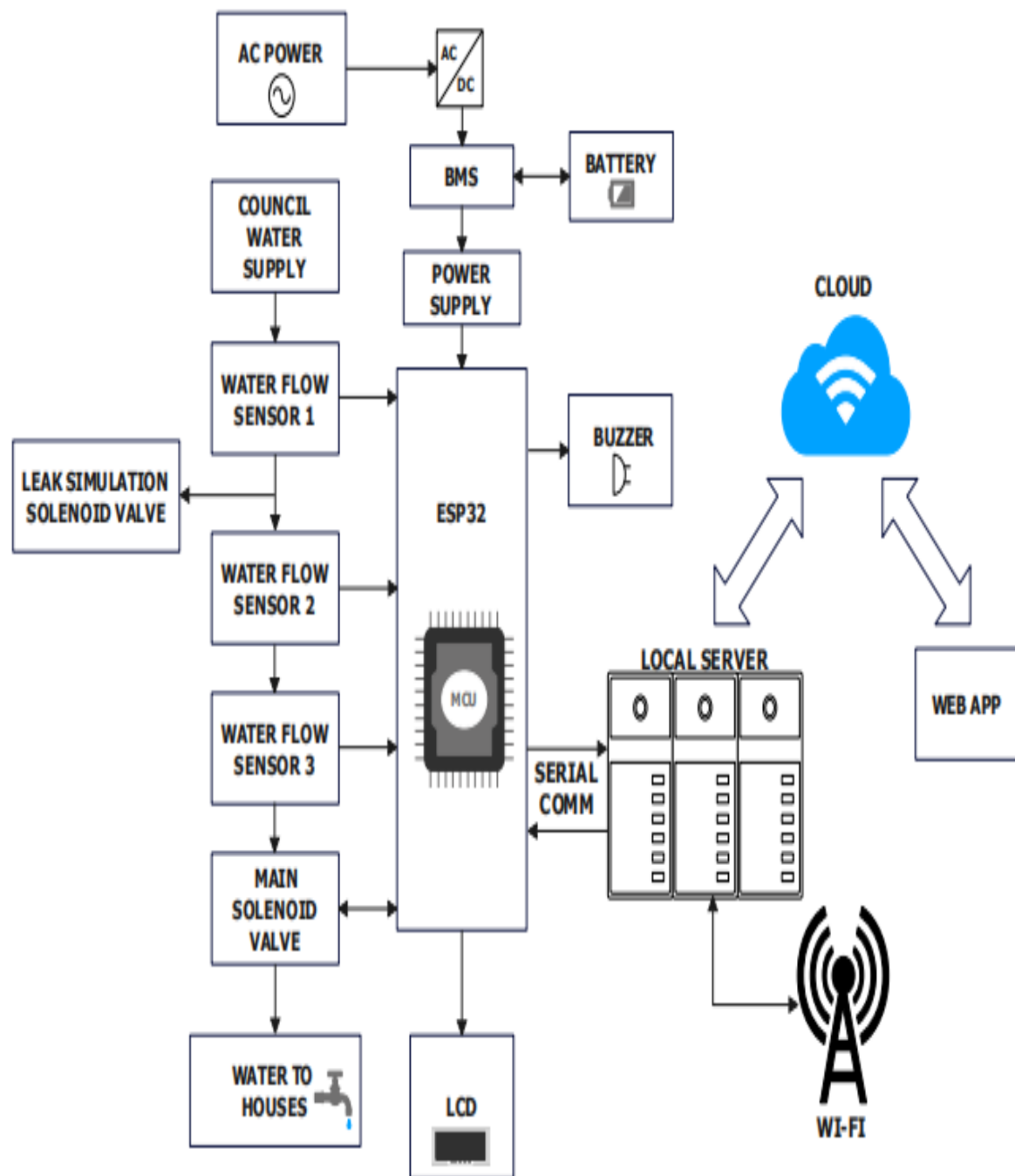


Figure 3. 18 Block Diagram

Subsystem Descriptions

The prepaid water meter is made up of connected subsystems. Each subsystem has a specific task, but the system only works correctly when sensing, processing, valve control, power supply, user display and backend communication operate together.

a) Power supply and battery backup subsystem

The power subsystem consists of the 12 V DC adapter, LM2596 buck converter, lithium battery pack and Battery Management System. The adapter supplies the main operating power, while the converter steps the voltage down for lower-voltage electronics. The battery pack provides backup during power cuts and the BMS protects the cells from unsafe charging or discharging.

b) Sensing Subsystem

The sensing subsystem uses three flow sensors. Two sensors are used on the simulated distribution line for leak detection, while the third measures household consumption. Each sensor produces pulse signals that are read by the ESP32 and converted into flow-rate and volume values.

c) Processing and Control Subsystem

The ESP32 is the main controller. It receives the pulse signals, calculates flow and volume, checks prepaid credit, manages leak-detection logic, controls the relay and sends system information to the backend through serial communication.

d) Actuation Subsystem

The actuation subsystem is based on the solenoid valve. The valve opens when the user has sufficient credit or approved emergency credit, and it closes when the system detects a zero-balance condition without available emergency credit.

e) Communication Subsystem

The communication subsystem uses serial communication between the ESP32 and the local server. The local server processes incoming readings, stores them in the database and is exposed for remote prototype access using Ngrok.

f) User Interface Subsystem

The local user interface consists of the 16 x 2 LCD and buzzer. The LCD displays balance, flow and system status, while the buzzer gives audible warnings for events such as low credit, zero balance or leak conditions.

g) Server and Web Application Subsystem

The web subsystem runs on the laptop server. It receives hardware data from the ESP32, updates the database, supports resident and council dashboards and connects to the Paynow payment process.

h) Water Distribution Subsystem

The water distribution subsystem consists of the tank, pipe fittings, sensors, valve and tap arrangement used to simulate the movement of water from a source to the consumer point.

Signal Flow Explanation

1. System Initialization and Power Supply

The system is powered by a 12 V DC adapter and supported by the backup battery pack. When the system starts, the ESP32 initialises the flow sensors, LCD, relay and buzzer, then checks whether valid sensor readings and balance data are available.

2. Water Flow Monitoring and Data Collection

When the valve is open, water flows through the distribution pipeline and household connection.

The system uses three flow sensors:

- Flow Sensor 1 (FS1) installed on the main water distribution pipeline.
- Flow Sensor 2 (FS2) positioned 500 m downstream from Flow Sensor 1 on the same distribution line.
- Flow Sensor 3 (FS3) installed at the consumer household connection point.

The flow sensors generate pulse signals proportional to the water flow passing through them.

These signals are continuously transmitted to the ESP32 for processing.

3. Leakage Detection and Localization within the Distribution Line

The ESP32 continuously compares the flow rates measured by Flow Sensor 1 and Flow Sensor 2 to detect leakage between the two points.

Leak detection logic:

- If $FS1 \approx FS2$, the system assumes normal operation and no leakage exists.
- If $FS1 > FS2$ beyond a predefined threshold, the system identifies a leakage occurrence along the pipeline section between the two sensors.
- Placing the two distribution sensors 500 m apart allows the system to identify the monitored section where a leak or burst is likely to be located. In a real council deployment, this would help maintenance teams narrow the search area before going to site.

When leakage is detected:

- Leakage alerts are immediately sent to the Council Dashboard
- Leak events are stored in the database
- Leakage status is displayed for administrators

Leak alerts are only available to the council administration dashboard and are not visible to residents.

4. Household Consumption Monitoring and Credit Calculation

Flow Sensor 3 continuously measures the amount of water entering the consumer's household.

The pulse signals generated are:

- Counted and converted into water volume (litres)
- Accumulated to determine total consumption
- Used to calculate remaining prepaid credit
- Used to calculate hourly, daily and monthly consumption records

The system continuously deducts consumed water from the available prepaid balance.

5. Credit Verification and Water Supply Control

The ESP32 continuously compares household water consumption with the remaining prepaid balance.

Decision process:

- If sufficient credit exists → continue water supply
- If credit becomes low → send low credit alert email and activate buzzer warning
- If credit reaches zero balance → send email notification, activate continuous buzzer alarm and automatically close the solenoid valve

Once the consumer tops up credits successfully:

- The backend updates the user's balance automatically
- The ESP32 synchronizes with the updated balance
- The solenoid valve automatically opens again
- Water supply resumes without manual intervention

6. Emergency Credit Functionality

The system provides an emergency credit request option through the web application.

The emergency credit process works as follows:

- Residents submit an emergency credit request through the system
- The council administrator reviews the request and either approves or rejects it
- If approved, temporary water access is granted

Emergency credit is treated as temporary support. It must be recovered on the next top-up, and the system prevents another emergency request until the previous one has been cleared.

7. Solenoid Valve Control

Based on the control logic, the ESP32 sends control signals to the relay module which controls the solenoid valve:

- Open valve → water supply continues
- Closed valve → water supply stops

The valve operation is therefore fully automated according to prepaid balance status.

8. User Interface and Notifications

The LCD display continuously shows:

- Remaining prepaid balance
- Water flow rate
- Water consumption values
- System status

The buzzer provides audible indications for:

- System initialization/startup
- Low credit warnings
- Zero balance alerts

Email notifications are sent for:

- System startup or wake-up events
- Low credit warnings
- Zero balance notifications

9. Data Transmission and Backend Processing

The ESP32 transmits system information through serial communication to the local server.

The transmitted information includes:

- Real-time flow rate readings
- Household water consumption
- Remaining balance
- Leakage status
- Solenoid valve status

The Python backend performs the following functions:

- Receive and validate incoming data
- Store data securely in the database
- Maintain cumulative consumption records
- Store billing history
- Track hourly, daily and monthly consumption
- Store leak event information

10. Web Application Functionality

i) Resident Interface

Residents can:

- Monitor water consumption in real time
- View hourly, daily and monthly usage history
- Download consumption reports in CSV format
- Receive balance-related notifications
- Purchase water credits through Paynow integration
- Request emergency credit

ii) Council Dashboard

Administrators can:

- Remotely manage consumer meters
- Monitor water usage
- Track distribution line leakage events
- View leak locations within monitored sections
- Generate reports for analysis and billing
- Approve or reject emergency credit requests

11. Payment Processing and Synchronization

When a user purchases water credits, payment is processed using Paynow integration. The backend automatically validates the transaction and updates the credit balance in real-time. The ESP32 synchronizes with the updated information immediately. If water supply had been stopped because of insufficient credit, the solenoid valve automatically opens and normal operation resumes.

Ngrok exposes the local server to the internet by:

- Creating a secure public URL
- Allowing remote access to the web application
- Simulating cloud deployment

12. Final System Output

The final system provides controlled water delivery with real-time monitoring, automated leak detection and prepaid billing functionality. Water is supplied only when sufficient credit or approved emergency credit exists. Leakage events occurring within the monitored distribution line are detected and reported to the council for rapid maintenance response, reducing treated water losses and improving overall water management efficiency.

3.5 System Flow Chart

The system flow chart illustrates the complete operational workflow of the Smart Water Meter with Tiered Pricing System. This chart demonstrates the interaction between hardware components (ESP32/Arduino, flow sensors, solenoid valve), software backend (Flask server, MySQL database), and user interfaces (web dashboard, payment gateway).

The flowchart demonstrates real-time interaction between hardware and software components. The ESP32 microcontroller continuously monitors water flow, applies tiered pricing calculations, and maintains synchronization with the cloud database through the Flask API. User actions such as balance top-ups immediately trigger valve state changes, ensuring seamless operation.

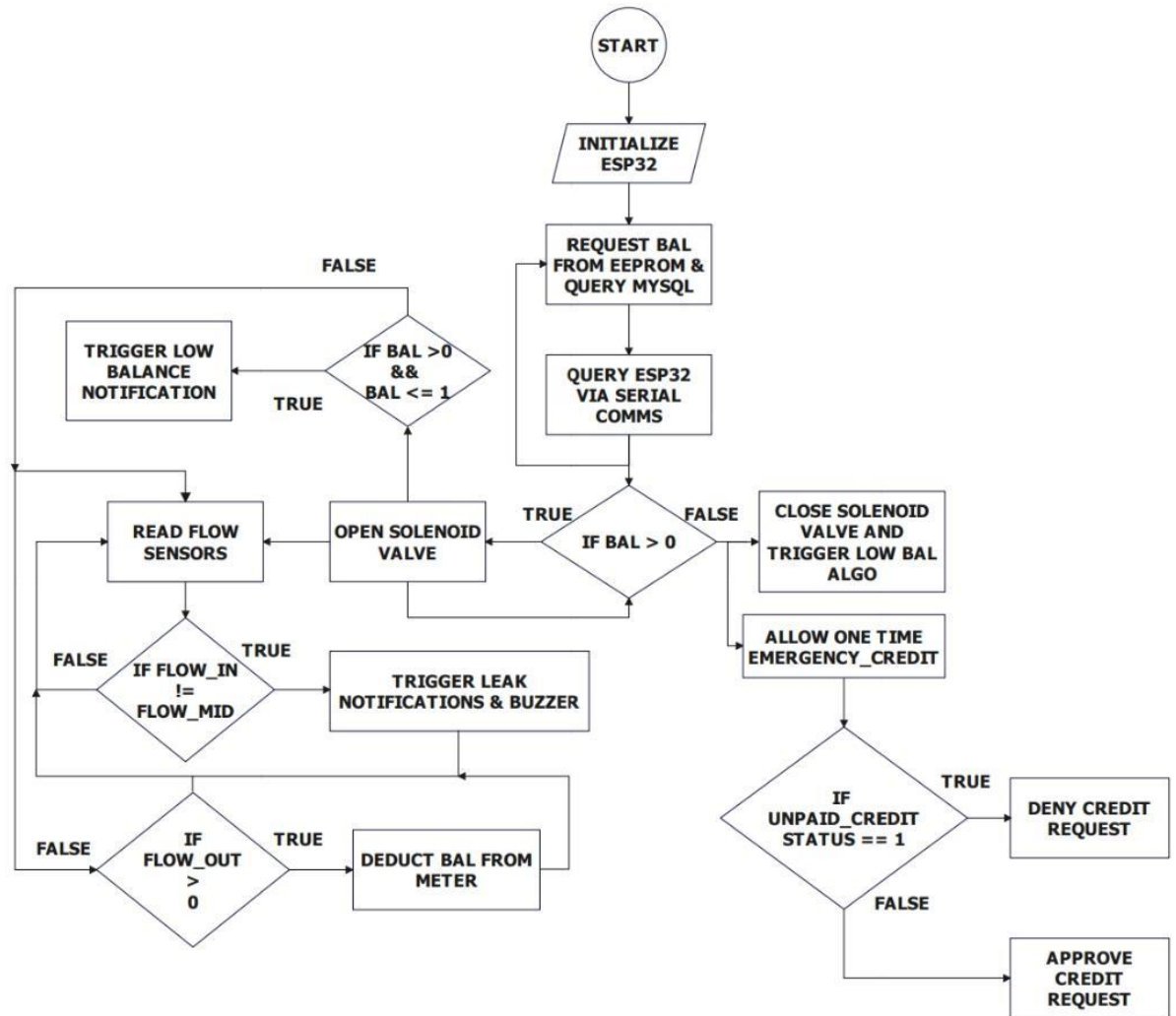


Figure 3. 19 Flow chart

3.6 System Design and Implementation

This section presents an overview of the design and implementation of the Smart Prepaid Water Metering system with real-time monitoring, leak detection and online payment integration,

covering hardware architecture, software framework, circuit design, algorithmic logic, and mathematical modeling. Each component has been engineered to work cohesively to deliver accurate water consumption tracking, real-time pricing calculations, and automated valve control.

Hardware Design

The hardware architecture consists of several integrated components that work together to monitor water consumption, control flow, and communicate with the backend system.

Core Components:

1. ESP32

Primary microcontroller responsible for sensor data acquisition, real-time calculations, valve control logic, and serial communication with the Flask backend. Operating at 240MHz with dual-core processing, it handles concurrent tasks including flow monitoring, LCD updates, and network communication.

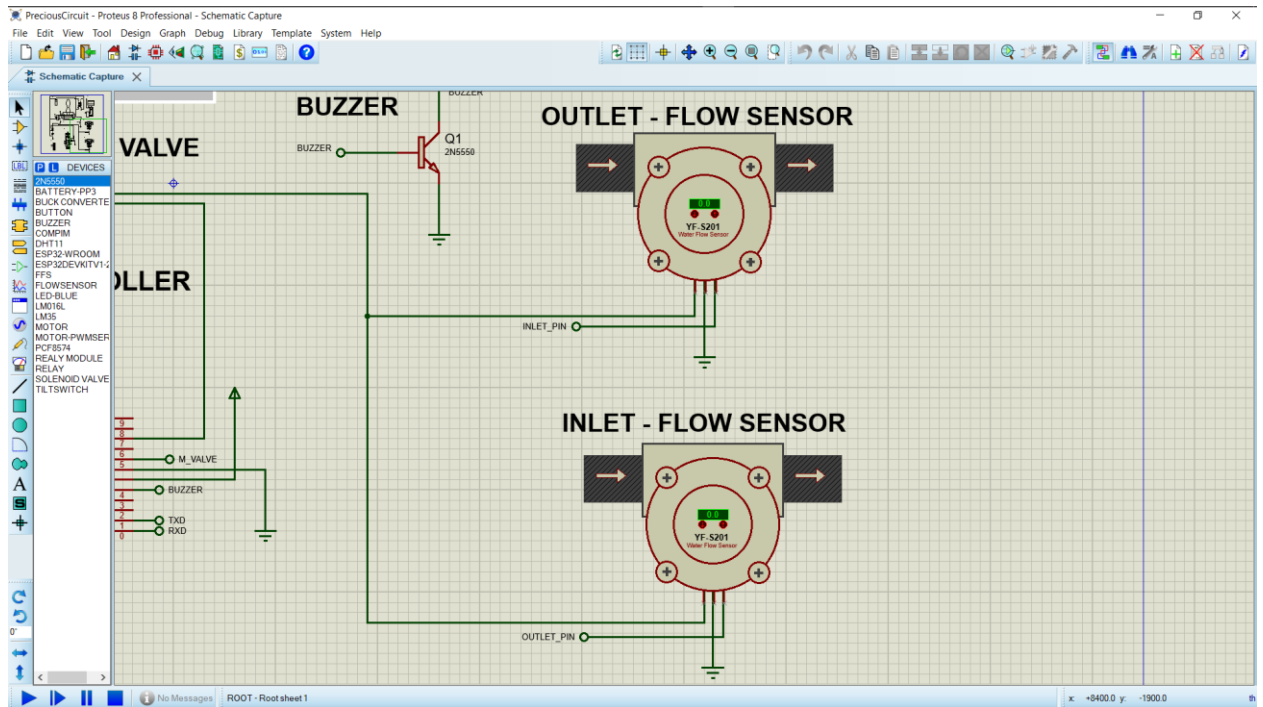


Figure 3. 21 Interfacing flow sensors

3. Solenoid Valve

12V DC normally-closed solenoid valve controlled via relay module connected to digital pin 5. Automatically opens when sufficient balance is available and closes when balance is depleted. Response time <50ms ensures immediate shutoff upon balance depletion. Rated for continuous operation with 200,000+ cycle lifespan.

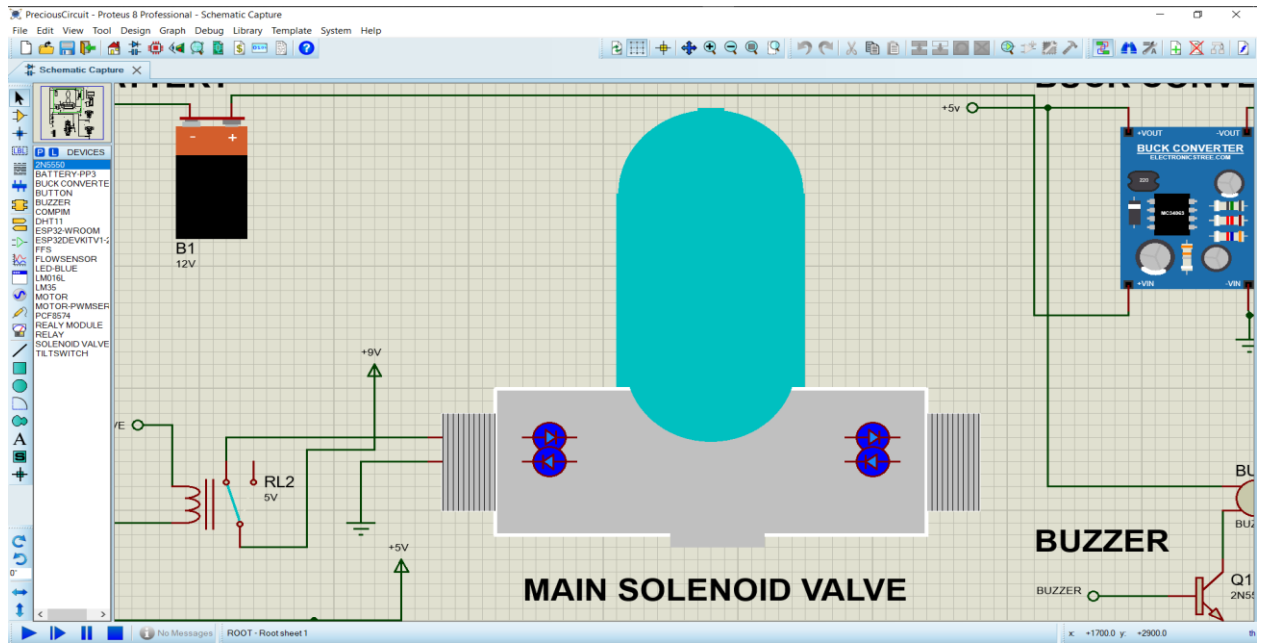


Figure 3. 22 Interfacing solenoid valve

4. LCD Display

16x2 character I2C LCD display (address 0x20) for real-time feedback. Displays current balance, flow rate, valve status, and leak alerts. I2C communication requires only 2 wires (SDA, SCL) reducing pin usage. Auto-brightness adjustment and backlight control for visibility in various lighting conditions.

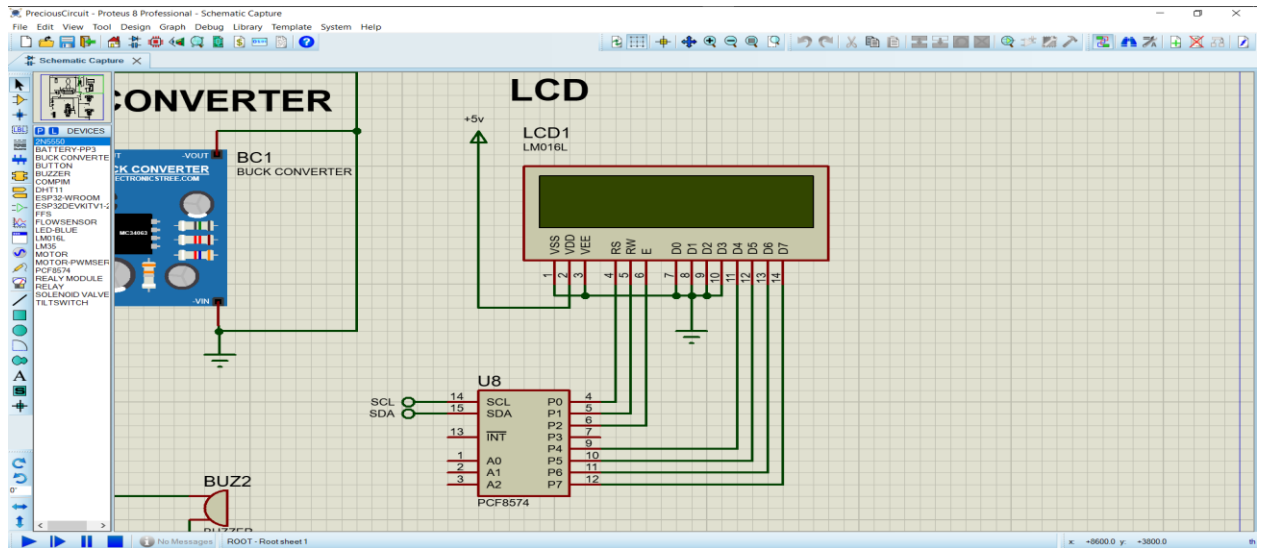


Figure 3. 23 Interfacing LCD

5. Buzzer

Active buzzer module on pin 4 for audible alerts. Provides differentiated beep patterns: single beep for valve open, double beep for valve closure, triple beep for balance depletion, and rapid beeps for leak detection alert so that residents can alert the council in case the council does not see the leakage notification on time. Operates at 85dB for indoor audibility.

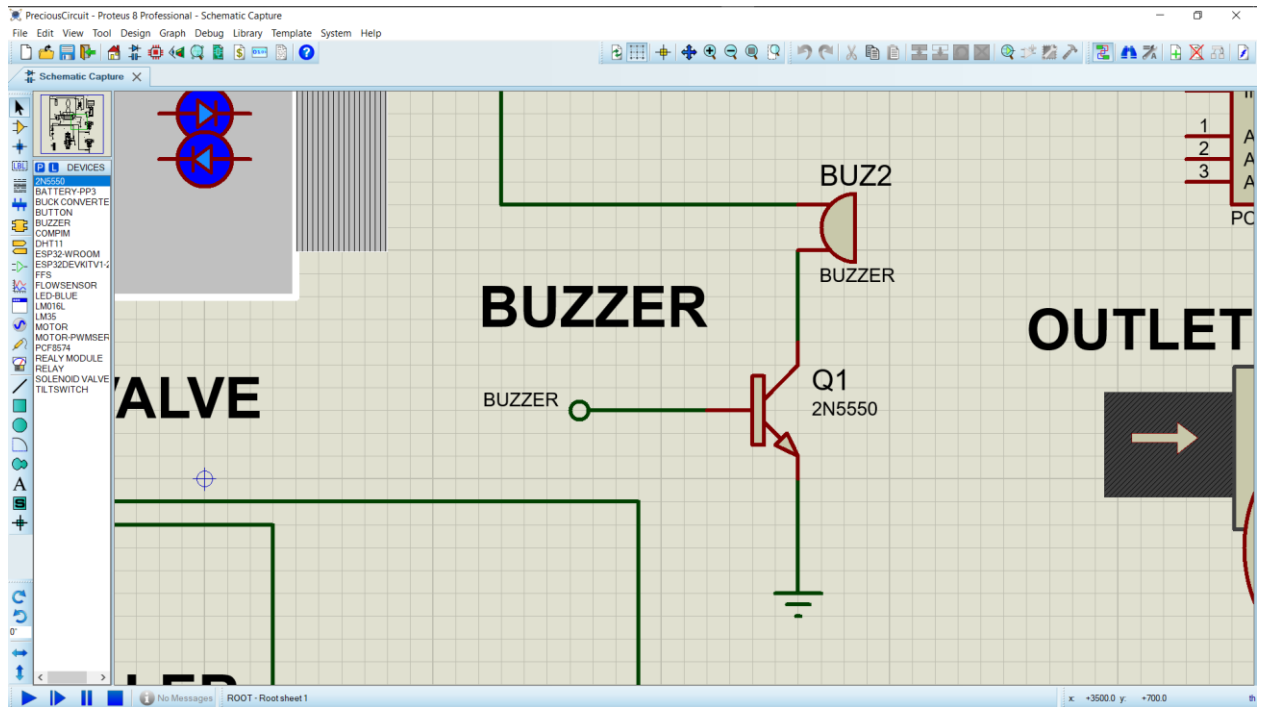


Figure 3. 24 Adding buzzer

6. Power Supply

Dual-voltage power system: 5V DC for ESP32 and sensors via USB/buck converter, 12V DC for solenoid valve via dedicated power adapter. Includes voltage regulation and surge protection circuits. Battery backup option (12V 7Ah) for 4-6 hours of operation during power outages.

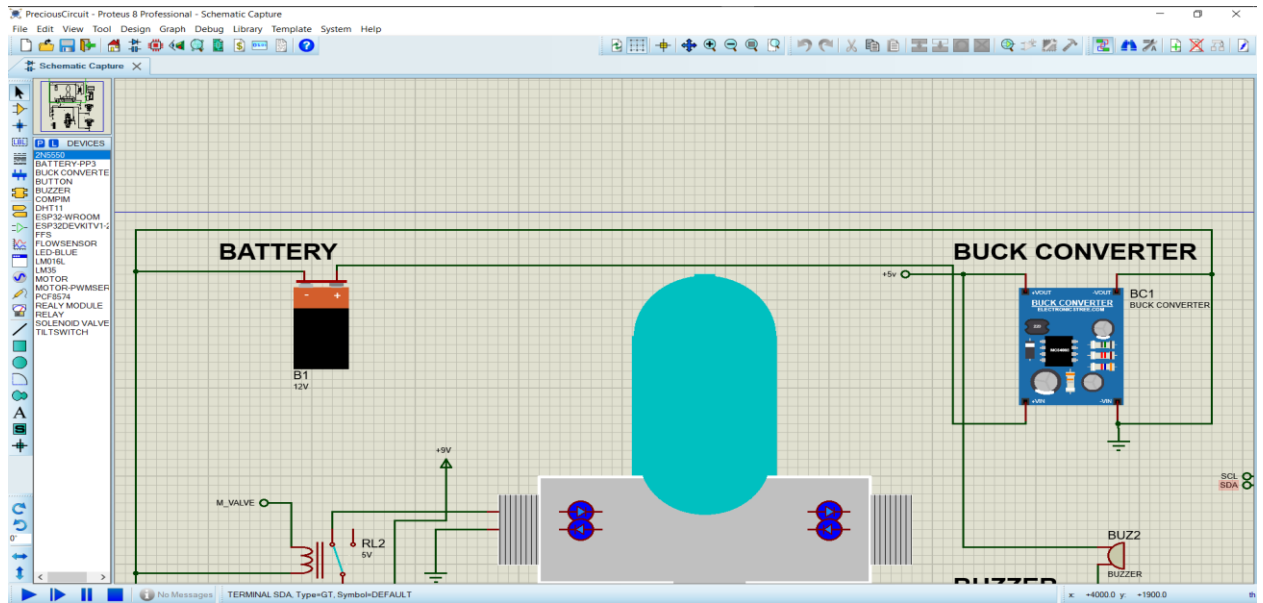


Figure 3. 25 Power supply

Software Design

The software architecture follows a three-tier model comprising embedded firmware, backend server logic, and web-based user interfaces. This modular design ensures scalability, maintainability, and real-time responsiveness.

Software Stack Overview:

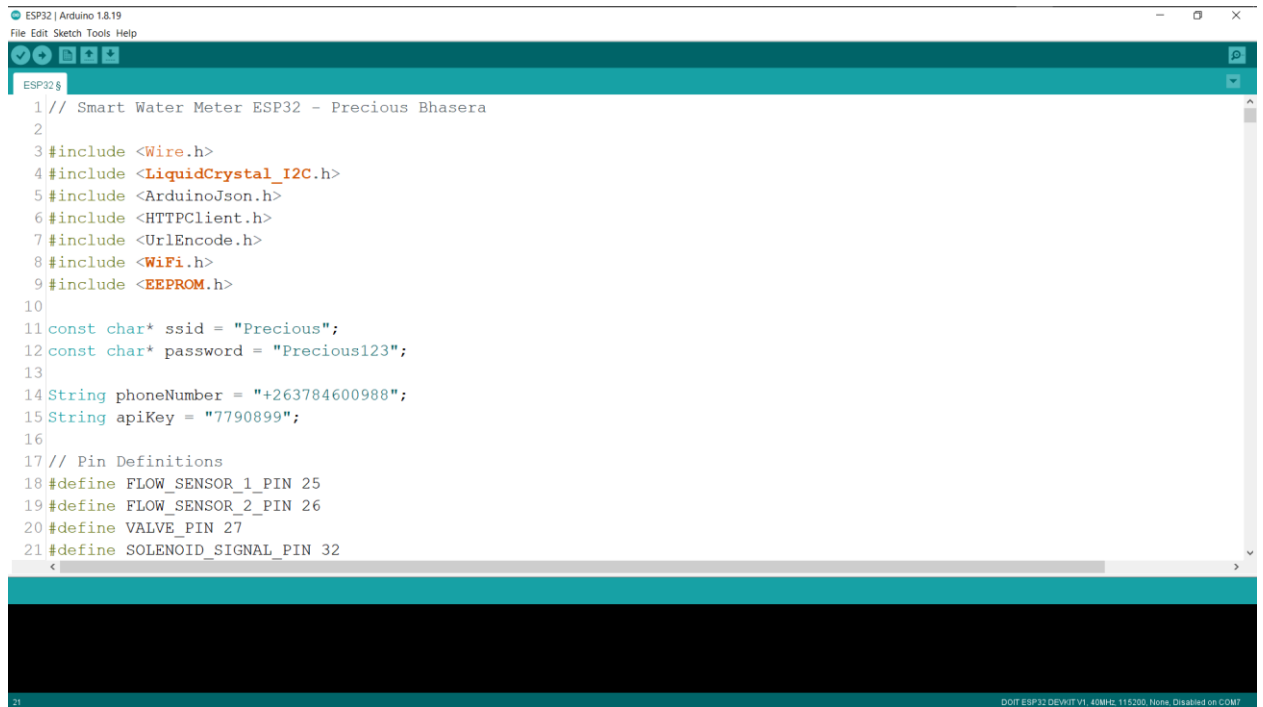
1. Firmware (Embedded)

Language: C++ (Arduino IDE)

Libraries: ArduinoJson (v6.x), LiquidCrystal_I2C, Wire

Key Functions: calculateFlow(), updateLCD(), handleSerialCommand(), calculateTieredCost()

Implements non-blocking loops with millis() timing for concurrent sensor monitoring, display updates (1s interval), and serial communication (5s interval). Utilizes JSON protocol for structured data exchange with backend.



```
1 // Smart Water Meter ESP32 - Precious Bhasera
2
3 #include <Wire.h>
4 #include <LiquidCrystal_I2C.h>
5 #include <ArduinoJson.h>
6 #include <HTTPClient.h>
7 #include <UriEncode.h>
8 #include <WiFi.h>
9 #include <EEPROM.h>
10
11 const char* ssid = "Precious";
12 const char* password = "Precious123";
13
14 String phoneNumber = "+263784600988";
15 String apiKey = "7790899";
16
17 // Pin Definitions
18 #define FLOW_SENSOR_1_PIN 25
19 #define FLOW_SENSOR_2_PIN 26
20 #define VALVE_PIN 27
21 #define SOLENOID_SIGNAL_PIN 32
```

Figure 3. 26 Firmware

2. Backend API

Framework: Flask (Python 3.9+)

Dependencies: Flask-CORS, PySerial, mysql-connector-python, requests

Key Endpoints: /api/user/login, /api/user/top-up, /api/pricing/tiers, /api/user/monthly-usage

RESTful API architecture with JWT authentication. Handles serial communication thread for ESP32 data reception, processes PayFast payment callbacks, and executes tiered pricing algorithms. Email notifications via SMTP for low balance and leak alerts.

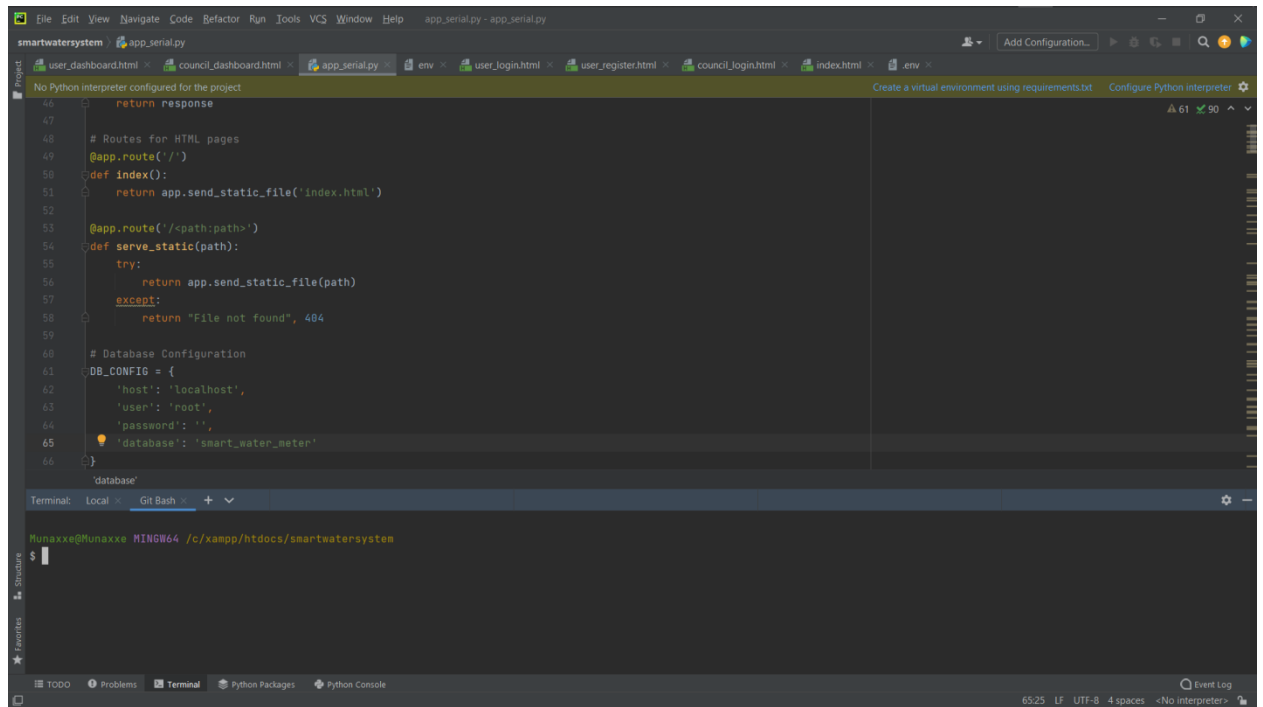


Figure 3. 27 Backend API

3. Database

DBMS: MySQL 8.0

Tables: users, meters, meter_readings, water_pricing_tiers, monthly_consumption_tracking, transactions, alerts

Stored Procedures: calculate_water_cost()

Relational schema with foreign key constraints ensures data integrity. Indexed columns (meter_number, user_id) optimize query performance. Event scheduler automates monthly consumption reset on 1st of each month. Trigger functions archive historical data to monthly_consumption_tracking table.

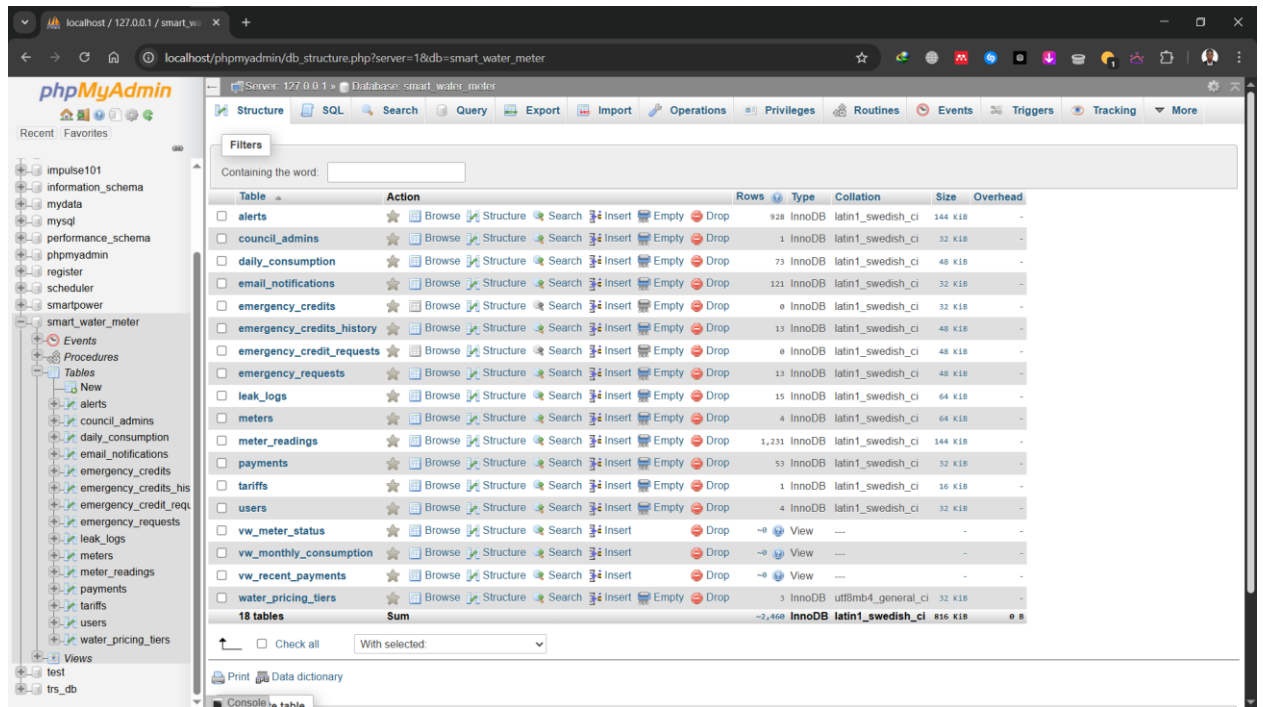


Figure 3. 28 Database

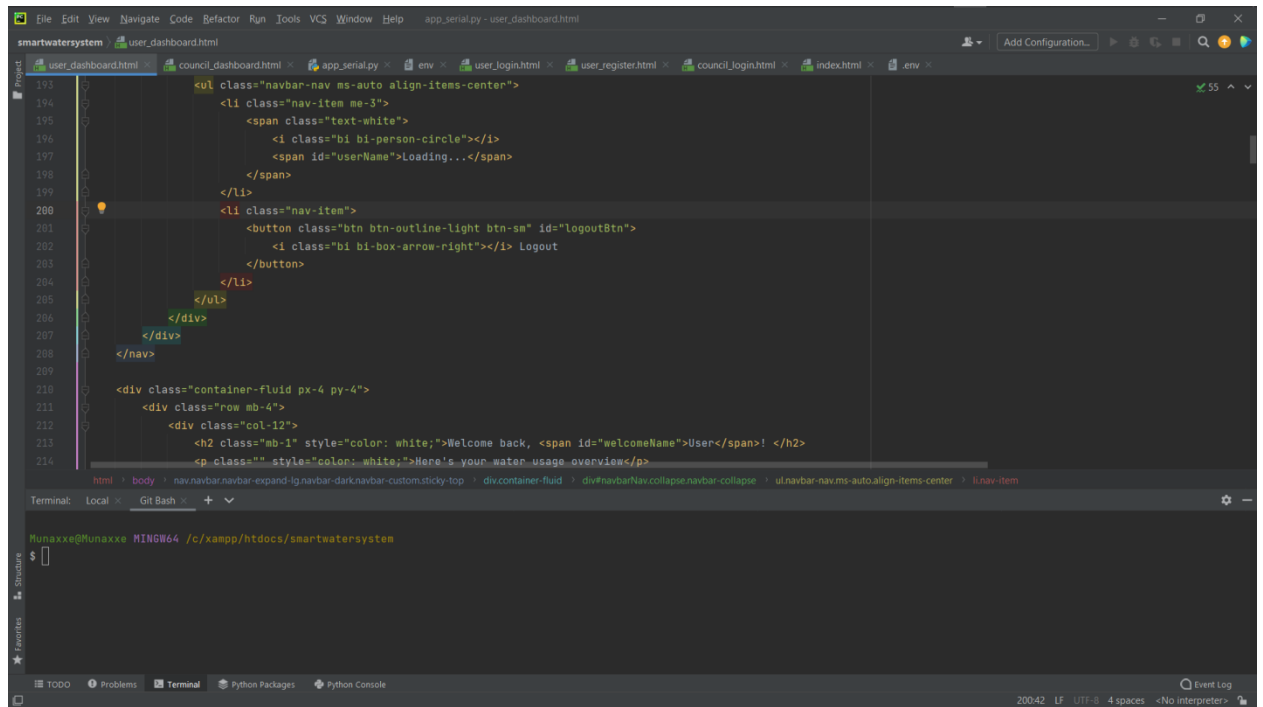
4. Frontend UI

Technologies: HTML5, CSS3, JavaScript (ES6+), Bootstrap 5.3

Pages: user_login.html, user_dashboard.html, council_login.html, council_dashboard.html

Key Scripts: tiered_pricing.js, dashboard.js

Responsive single-page applications with asynchronous API calls via Fetch API. Real-time balance updates, consumption visualizations with Chart.js, and interactive pricing breakdown modals. LocalStorage for JWT token persistence. Progressive Web App (PWA) capabilities for offline access to historical data.



```
193     <ul class="navbar-nav ms-auto align-items-center">
194       <li class="nav-item me-3">
195         <span class="text-white">
196           <i class="bi bi-person-circle"></i>
197           <span id="userName">Loading...</span>
198         </span>
199       </li>
200       <li class="nav-item">
201         <button class="btn btn-outline-light btn-sm" id="logoutBtn">
202           <i class="bi bi-box-arrow-right"></i> Logout
203         </button>
204       </li>
205     </ul>
206   </div>
207 </div>
208 </nav>
209
210 <div class="container-fluid px-4 py-4">
211   <div class="row mb-4">
212     <div class="col-12">
213       <h2 class="mb-1" style="color: white;">Welcome back, <span id="welcomeName">User</span>! </h2>
214       <p class="" style="color: white;">Here's your water usage overview</p>
215     </div>
216   </div>
217 </div>
```

Figure 3. 29 Frontend UI

Circuit Design

The circuit design integrates multiple subsystems including sensor interfaces, valve control, display modules, and communication peripherals. All connections follow standard electrical safety practices with appropriate current limiting resistors and voltage regulation.

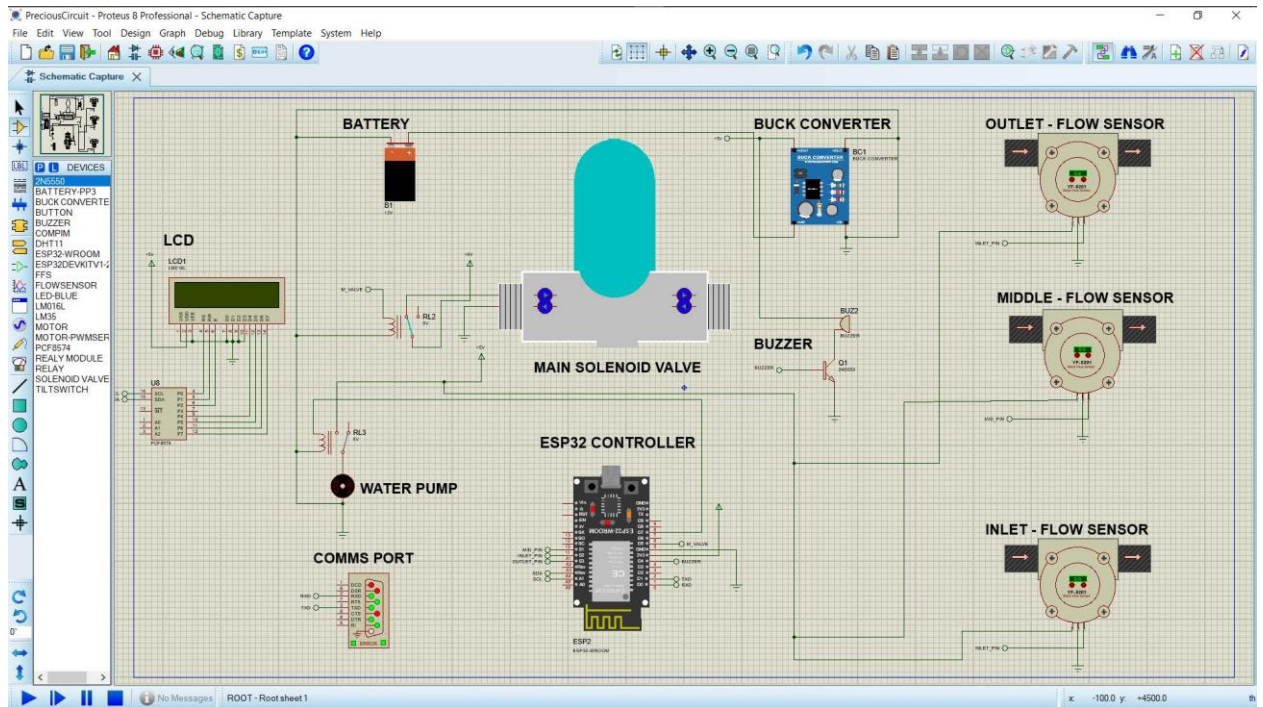


Figure 3. 30 Overall circuit diagram

Table 3. 2 Circuit Components and Connections

Connection	Pin/Terminal	Technical Specifications
Flow Sensor 1 (Inlet)	25	5V power, GND, signal via 10kΩ pull-up resistor. 0-5V analog output.
Flow Sensor 2 (Outlet)	26	5V power, GND, signal via 10kΩ pull-up resistor. 0-5V analog output.
Solenoid Valve Relay	27	5V relay module, 10A contact rating. Flyback diode (1N4007) for inductive load protection.

Buzzer	33	Active buzzer 5V, 30mA. NPN transistor (2N2222) driver with 1k Ω base resistor.
LCD Display (I2C)	SDA, SCL	I2C address 0x20, 4.7k Ω pull-up resistors on SDA/SCL lines. 5V logic.
Serial Communication	TX, RX	9600 baud UART. USB-to-Serial adapter (CH340/CP2102) for PC connection.
Power Regulation	VIN, GND	LM7805 5V regulator (1A), 12V input. 100 μ F + 0.1 μ F bypass capacitors.

Algorithms And Pseudo Code

The system employs several key algorithms for water consumption monitoring, tiered pricing calculation, and automated control. Below is the core algorithms implemented in the system.

Algorithm 1: Flow Rate Calculation and Consumption Tracking

ALGORITHM: calculateFlow()

INPUT: Analog sensor readings from flow sensors

OUTPUT: Flow rate (L/min), total volume consumed (L)

```

BEGIN  IF balance_not_initialized OR valve_closed THEN    RETURN // Skip calculation  END
IF      // Read sensor values (0-1023)  inlet_raw ← READ_ANALOG(A1)  outlet_raw ←
READ_ANALOG(A2)      // Convert to flow rate (L/min)  // Formula: FlowRate = (SensorValue /
1024) × MaxFlow  flow_inlet ← (inlet_raw / 1024.0) × 10.0  flow_outlet ← (outlet_raw / 1024.0) ×
10.0      // Detect leaks by comparing inlet vs outlet  flow_difference ← ABS(flow_inlet - flow_outlet)
IF flow_difference > 0.5 THEN    leak_detected ← TRUE    TRIGGER_BUZZER(2 beeps) // Leak
alert  ELSE    leak_detected ← FALSE  END IF      // Use outlet flow for billing  IF flow_outlet
< MIN_FLOW_THRESHOLD THEN    flow_outlet ← 0 // Ignore noise    RETURN  END IF

```

```

// Calculate volume in this time interval // Volume (L) = FlowRate (L/min) × Time (sec) / 60
interval_time ← 1.0 // 1 second reading interval volume ← (flow_outlet / 60.0) × interval_time
total_volume ← total_volume + volume // Calculate cost using tiered pricing cost ←
CALCULATE_TIERED_COST(volume, current_month_consumption) // Deduct from balance IF
current_balance > cost THEN current_balance ← current_balance - cost ELSE
current_balance ← 0 CLOSE_VALVE() TRIGGER_BUZZER(3 beeps) // Balance depleted alert
SEND_EMAIL_ALERT("Balance Depleted") END IF // Update LCD display
UPDATE_LCD(current_balance, flow_outlet, valve_status) END

```

Algorithm 2: Tiered Pricing Calculation

ALGORITHM: calculateTieredCost()

INPUT: liters_consumed, current_month_consumption

OUTPUT: total_cost, tier_breakdown BEGIN

```

// Initialize variables new_total_consumption ← current_month_consumption + liters_consumed
remaining_liters ← liters_consumed tier1_liters ← 0 tier2_liters ← 0 tier3_liters ← 0
tier1_cost ← 0 tier2_cost ← 0 tier3_cost ← 0 // Tier 1: 0-1000L @ $0.005/L tier1_max ←
1000 tier1_rate ← 0.005 IF current_month_consumption < tier1_max THEN tier1_available
← tier1_max - current_month_consumption tier1_liters ← MIN(remaining_liters, tier1_available)
tier1_cost ← tier1_liters × tier1_rate remaining_liters ← remaining_liters - tier1_liters END IF
// Tier 2: 1001-5000L @ $0.010/L tier2_max ← 5000 tier2_rate ← 0.010 IF remaining_liters
> 0 AND new_total_consumption > tier1_max THEN tier2_start ←
MAX(current_month_consumption, tier1_max) IF tier2_start < tier2_max THEN
tier2_available ← tier2_max - tier2_start tier2_liters ← MIN(remaining_liters, tier2_available)
tier2_cost ← tier2_liters × tier2_rate remaining_liters ← remaining_liters - tier2_liters END
IF END IF // Tier 3: 5001+L @ $0.050/L tier3_rate ← 0.050 IF remaining_liters > 0
THEN tier3_liters ← remaining_liters tier3_cost ← tier3_liters × tier3_rate END IF //
Determine current tier IF new_total_consumption <= 1000 THEN current_tier ← 1 ELSE IF
new_total_consumption <= 5000 THEN current_tier ← 2 ELSE current_tier ← 3 END IF
// Calculate total cost total_cost ← tier1_cost + tier2_cost + tier3_cost // Return breakdown
RETURN { total_cost: total_cost, tier1: {liters: tier1_liters, cost: tier1_cost}, tier2: {liters:

```

*tier2_liters, cost: tier2_cost}, tier3: {liters: tier3_liters, cost: tier3_cost}, current_tier:
current_tier, new_total_consumption: new_total_consumption } END*

Algorithm 3: Monthly Reset and Archive

ALGORITHM: checkAndResetMonthlyConsumption()

INPUT: meter_number

OUTPUT: reset_performed (boolean)

```
BEGIN   current_month ← GET_FIRST_DAY_OF_MONTH(CURRENT_DATE)      // Query meter
data from database   meter_data ← DATABASE.QUERY(      "SELECT current_month_consumption,
last_tier_reset      FROM meters WHERE meter_number = ?"      , meter_number)      last_reset
← meter_data.last_tier_reset   current_consumption ← meter_data.current_month_consumption      //
Check if month has changed   IF last_reset < current_month THEN      // Archive previous month's
data      IF current_consumption > 0 THEN      DATABASE.EXECUTE(      "INSERT INTO
monthly_consumption_tracking      (meter_number, billing_month, total_liters)
VALUES (?, ?, ?)",      meter_number, last_reset, current_consumption)      END IF      //
Reset consumption for new month      DATABASE.EXECUTE(      "UPDATE meters      SET
current_month_consumption = 0,      current_tier = 1,      last_tier_reset = ?
WHERE meter_number = ?",      current_month, meter_number)      // Log reset event
LOG("Monthly reset performed for " + meter_number)      RETURN TRUE   END IF
RETURN FALSE END
```

Mathematical Modelling

The system's functionality is governed by several mathematical models that ensure accurate measurement, fair pricing and reliable operation.

1. Flow Rate Conversion Model

The YF-S201 Hall effect flow sensor outputs a frequency proportional to flow rate. The relationship is:

$$\text{Flow Rate (L/min)} = (\text{Analog Reading} / 1024) \times \text{Maximum Flow Rate}$$

Where:

- Analog Reading: 10-bit ADC value (0-1023)
- Maximum Flow Rate: 10 L/min (sensor specification)
- Output Range: 0-10 L/min

Volume consumed over time interval Δt :

$$\text{Volume (L)} = \text{Flow Rate (L/min)} \times \Delta t (\text{seconds}) / 60$$

2. Tiered Pricing Model

The cost function $C(V)$ for volume V consumed in a month is piecewise:

$$C(V) = r_1 \times V$$

$$\begin{aligned} & \text{if } 0 \leq V \leq 1000 \quad r_1 \times 1000 + r_2 \times (V - 1000) \quad \text{if } 1000 < V \leq 5000 \quad r_1 \times 1000 + r_2 \times 4000 + r_3 \times \\ & (V - 5000) \quad \text{if } V > 5000 \end{aligned}$$

Where:

- $r_1 = \$0.005/\text{L}$ (Tier 1 rate)
- $r_2 = \$0.010/\text{L}$ (Tier 2 rate)
- $r_3 = \$0.050/\text{L}$ (Tier 3 rate)

Table 3. 3 Example calculations:

Volume (L)	Calculation	Total Cost
500	500×0.005	\$2.50
2500	$1000 \times 0.005 + 1500 \times 0.010$	\$20.00
7000	$1000 \times 0.005 + 4000 \times 0.010 + 2000 \times 0.050$	\$145.00

3. Leak Detection Model

Leak detection is based on the flow differential between inlet and outlet sensors:

$$Leak\ Detected = |Flow\ inlet - Flow\ outlet| > Threshold$$

Where Threshold = 0.5 L/min (configurable parameter)

Leak severity classification:

- Minor leak: 0.5 - 1.0 L/min difference (warning alert)
- Moderate leak: 1.0 - 3.0 L/min difference (email notification)
- Severe leak: > 3.0 L/min difference (auto valve closure)

4. Balance Depletion Rate Model

The rate of balance depletion depends on consumption rate and current pricing tier:

$$\frac{dB}{dt} = -ri \times Q$$

Where:

- dB/dt = rate of balance change (\$/min)
- r_i = current tier rate (\$/L)
- Q = flow rate (L/min)

Time until balance depletion:

$$t_{empty} = B / (r_i \times Q)$$

This model enables predictive low-balance warnings to be sent before complete depletion.

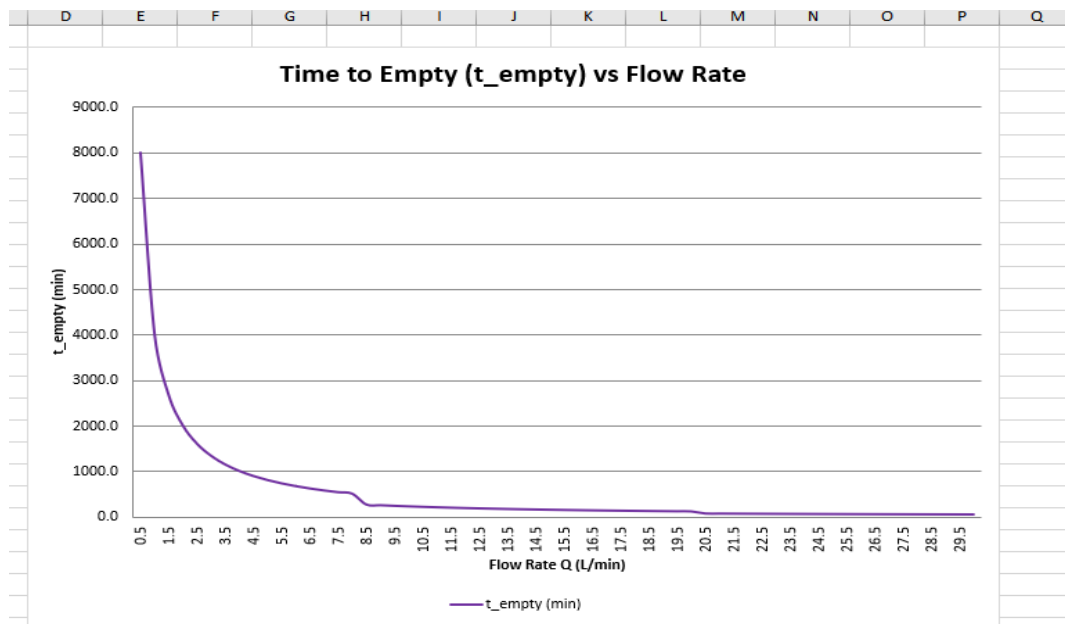


Figure 3. 31 Time to empty vs Flow rate

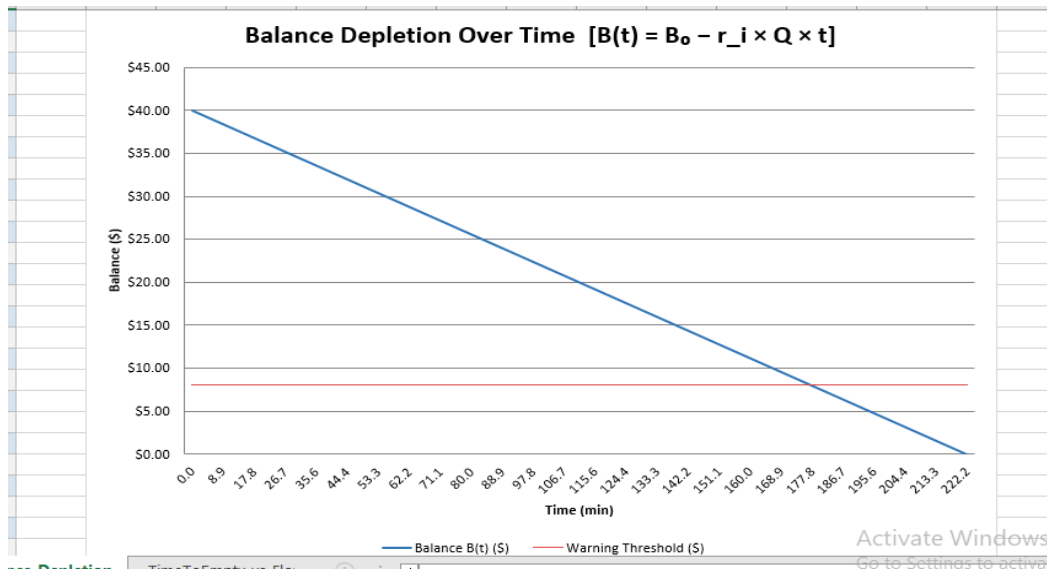


Figure 3. 32 Balance depletion over time

3.8 Ethical Considerations

The deployment of the Smart Prepaid Water Meter with Real-Time Monitoring, Leak Detection and Online Payment Integration raises ethical considerations to ensure fair, secure and responsible implementation.

1. User Privacy and Data Protection

The system collects and stores water consumption data and user account information through the web application.

- **Informed Consent:** Participants involved in testing must be informed about the system operation and data collection process before participation.
- **Data Protection:** User information and consumption records must be securely stored using authentication controls and encrypted communication channels.
- **Confidentiality:** Data collected during testing will only be used for academic and system evaluation purposes.
- **Right to Withdraw:** Participants can withdraw from testing at any stage without consequences.

2. Access to Water Services

Access to water is an essential human need; therefore, the system should avoid completely denying water access.

- **Emergency Credit Facility:** The system provides a one-time council-administered emergency credit feature allowing residents with depleted balances to request temporary water access.
- **Council Approval:** Emergency requests are reviewed and approved by council administrators to avoid misuse.
- **Continuous Service Consideration:** The emergency feature ensures households are not immediately left without access to water.

3. System Reliability and Safety

System failures may affect water supply and user confidence.

- **Battery Backup:** A battery backup system maintains operation during power outages.
- **Failsafe Operation:** The system should prevent unintended valve closure during communication or hardware failures.
- **Maintenance:** Regular monitoring and maintenance of sensors and valves are required to ensure reliable performance.

4. Integration with Existing Infrastructure

The system should support gradual implementation within existing council infrastructure.

- **Compatibility:** The proposed system should operate alongside existing water systems during deployment.
- **Installation Consent:** Property owners or users should be informed before installation.
- **Data Ownership:** Users should be able to access their water consumption records through the dashboard.

5. Environmental and Social Considerations

The system may have wider effects on society and the environment.

- **Water Conservation:** Real-time monitoring and leak detection reduce treated water losses and encourage responsible water use.
- **Electronic Waste:** Proper disposal and recycling of electronic components should be considered.
- **Accessibility:** Alternative methods such as SMS notifications can be introduced for users without internet access.

6. Regulatory Compliance

The system should comply with relevant regulations and standards.

- **Consumer Protection:** Users should have clear procedures for reporting billing or system issues.
- **System Accuracy:** Sensors should be calibrated and tested to ensure accurate measurements.
- **Security Compliance:** Data handling and storage should follow applicable privacy and data protection guidelines.

In summary, ethical use of the system requires more than technical performance. Privacy, fairness, reliability and access to water must be considered before any real deployment.

3.9 Summary

This chapter described the design, implementation and operation of the Smart Prepaid Water Meter with Real-Time Monitoring, Leak Detection and Online Payment Integration. It explained how the ESP32, flow sensors, solenoid valve, relay, LCD, buzzer, battery backup, Flask backend, database, Paynow integration and web dashboards were connected into one prototype. It also described the testing approach and ethical issues considered during development.

Chapter 4: EXPERIMENTAL RESULTS AND ANALYSIS

4.1 Introduction

This chapter presents what happened when the prototype was built and tested. The focus is on the evidence collected from the hardware, dashboards, payment process, valve control, leak detection and emergency-credit functions.

The earlier chapters explained the problem, reviewed related work and described the design. This chapter uses the completed prototype to show whether those design decisions produced a working system.

The chapter is organised into implementation results, data presentation, analysis and discussion. The implementation section shows the hardware and software outputs. The data section presents readings and test results. The analysis section explains what the results mean in relation to the project objectives.

Together, the tests show how the prototype behaved under controlled conditions. The results should be read as prototype evidence rather than proof that the system is ready for full municipal deployment.

4.2 System Implementation Results

The prototype was assembled and tested under controlled conditions. This section presents evidence from the hardware build, circuit design, web dashboards, payment process, leak alerts and emergency-credit workflow.

4.2.1 Hardware Prototype



Figure 4. 1 Hardware prototype

The hardware prototype was arranged to represent a small prepaid water distribution setup. It included a reservoir, three flow sensors, an ESP32, a relay, a 12 V solenoid valve, an LCD, a buzzer, a battery backup and pipe fittings.

During testing, water moved from the reservoir through the pipe network when the tap was opened. The first two flow sensors were used to compare flow along the simulated distribution line, while the third sensor measured the water reaching the household connection.

The flow sensors sent pulse signals to the ESP32. The controller converted those pulses into flow and volume readings and used the programmed tariff logic to update the prepaid balance.

The LCD gave local feedback by showing balance, flow and system messages. At the same time, the ESP32 sent readings to the Flask backend so that the dashboard could display current system information.

The solenoid valve was used as the main supply-control device. It stayed open when the user had credit and closed after the balance reached zero. This showed the basic prepaid control loop required by the project.

When a Paynow top-up was completed, the backend updated the user's balance and the ESP32 received the new value through the system logic. The relay then opened the valve again. Emergency credit was tested in a similar way, except that approval came from the council dashboard.

Leak detection was tested by comparing the two distribution flow readings. When the difference between the two readings exceeded the set threshold, the prototype treated the condition as a leak and produced alerts.

The battery backup was included to keep the prototype active during power interruptions. It allowed the controller, sensing and monitoring functions to continue when mains power was removed during testing.

The hardware tests confirmed that the main subsystems could operate together in one prototype: prepaid balance control, flow measurement, leak detection, emergency credit, online top-up, display alerts and backup power.

4.2.2 Circuit Design

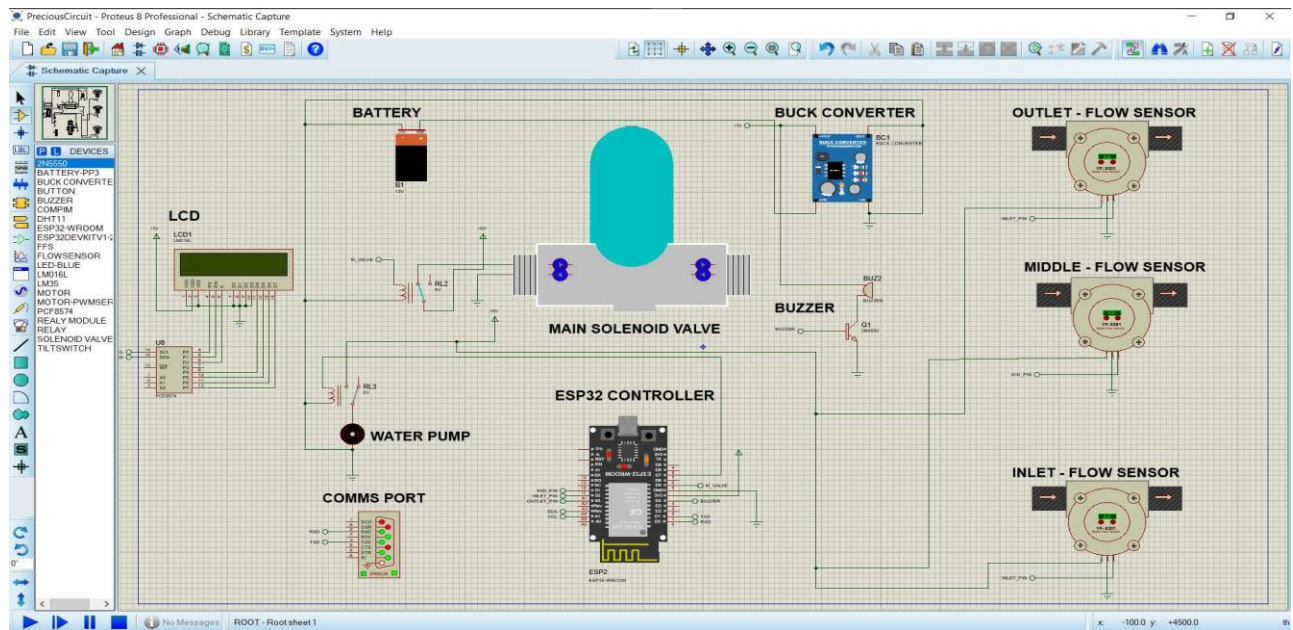


Figure 4. 2 Full circuit diagram

The circuit connects the sensing, processing, control, communication and power sections of the prototype. Each section has a clear function, but the system depends on correct interaction between them.

The power section starts with a 12 V DC adapter. The adapter supplies the solenoid valve and feeds the LM2596 converter, which provides lower voltages for the ESP32 and supporting electronics. The battery backup is connected through the BMS to protect the lithium cells.

Three flow sensors are used. The first sensor provides the upstream distribution reading, the second provides the downstream distribution reading for leak detection and the third measures household consumption.

The leak-detection function depends on comparing the first and second sensor readings. When the readings differ beyond the allowed threshold, the ESP32 identifies an abnormal condition in the monitored section.

The household sensor provides the consumption value used for prepaid deduction. This separates distribution-line monitoring from the user's actual metered consumption.

Each sensor produces pulses as water passes through it. The ESP32 counts the pulses and converts them into flow and volume values using the calibration factor in the firmware.

The prepaid billing logic reduces the user's balance as water is used. If the balance reaches zero, the controller closes the valve unless emergency credit has been approved.

The payment feature allows a user to recharge through the web dashboard. After confirmation, the backend updates the balance and the ESP32 uses the updated value to restore valve operation.

The emergency-credit feature gives temporary access when the balance is exhausted. The request is reviewed by the council administrator and is recorded so that repayment can be required during the next top-up.

The ESP32 sends readings and status information to the Flask backend. The dashboard then displays consumption, balance, leak status and valve state to the resident or council user depending on the account type.

The relay module allows the low-voltage ESP32 to switch the higher-current solenoid valve safely. The optocoupler provides isolation between the control and load sides of the circuit.

Under normal operation, the valve remains open. If credit is depleted and no emergency credit is active, the relay deactivates and the valve closes. When credit is restored, the relay activates again and the valve opens.

4.2.3 Resident Web Dashboard

The resident dashboard was built with HTML, CSS, JavaScript and Bootstrap and served through the Flask backend. It allows the resident to view current balance, consumption, payment history and emergency-credit options from one interface.

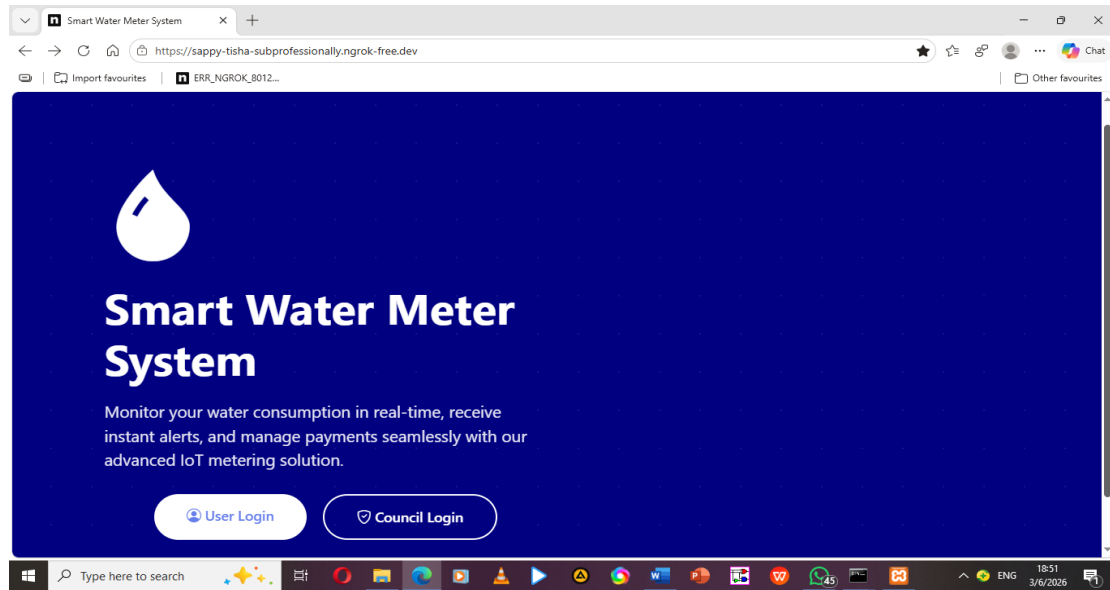


Figure 4. 3 Web dashboard

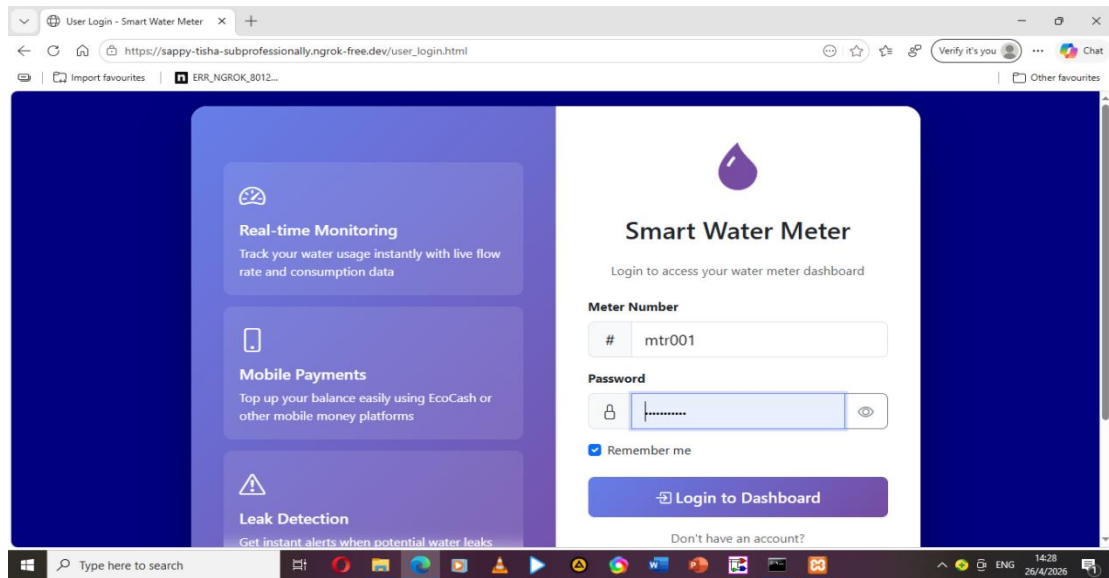


Figure 4. 4 Resident logging in

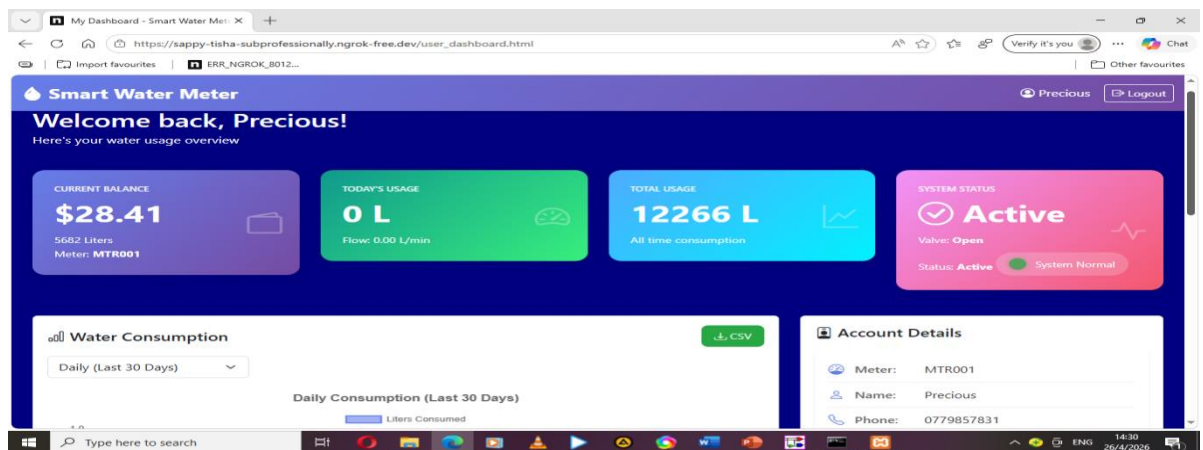


Figure 4. 5 Resident dashboard

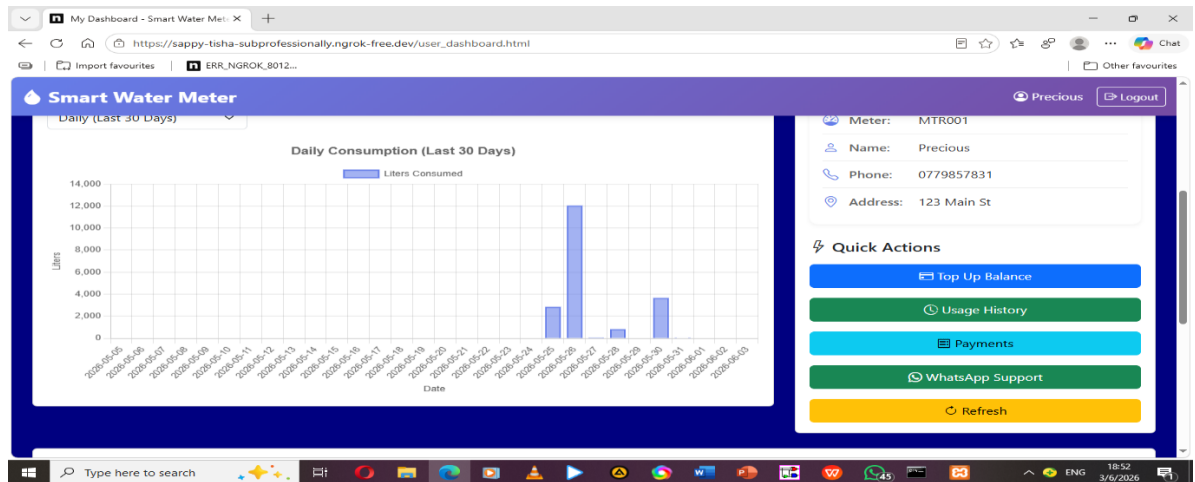


Figure 4. 6 Resident web dashboard

4.2.4 Council Administrator Dashboard

The council dashboard gives administrators a view of registered meters, flow readings, leak alerts, payment records and emergency-credit requests. It is intended to support monitoring and decision-making rather than replace full council billing systems at this prototype stage.

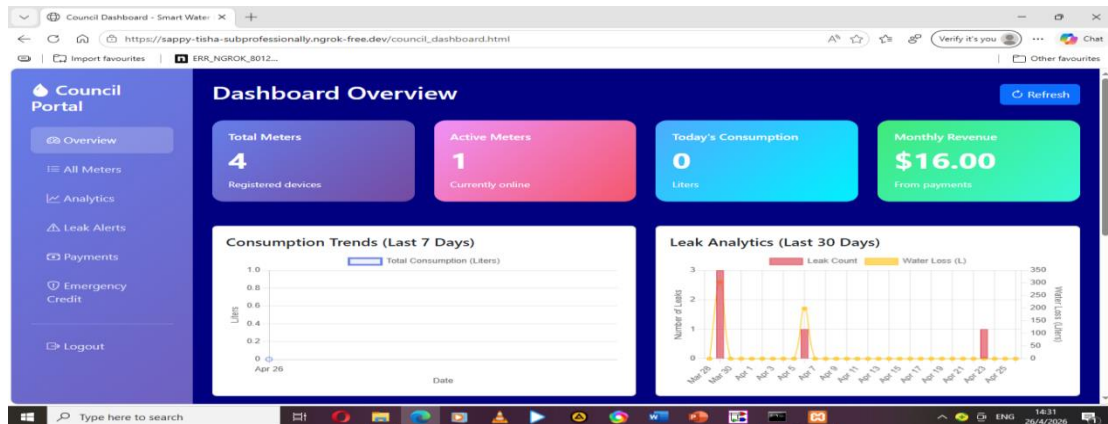


Figure 4. 7 Council dashboard

4.2.5 Online Payment Integration

When a resident starts a top-up, the Flask backend sends the payment request to Paynow. After payment confirmation, the backend updates the user's balance in the database and the new balance is used by the ESP32 control logic.

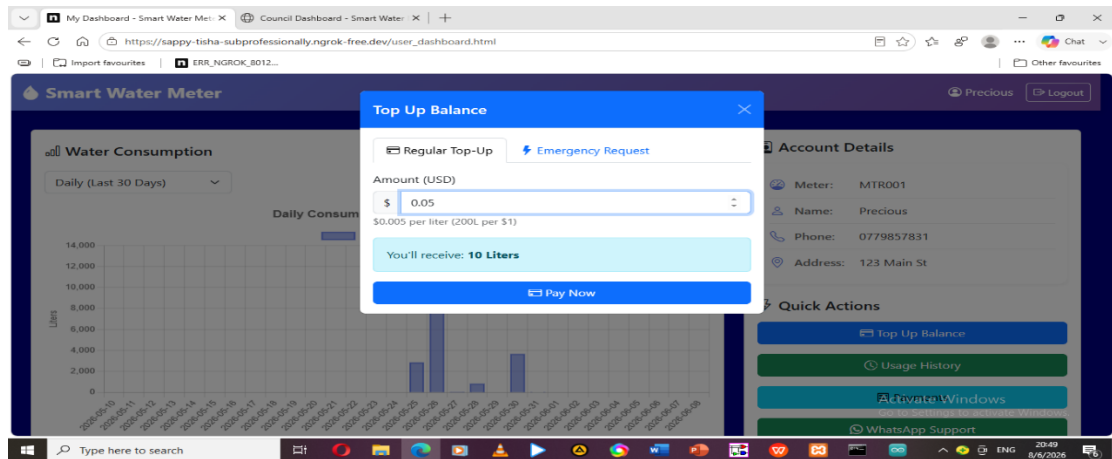


Figure 4. 8 Online payment request

The payment page redirects the user to Paynow so that the test payment can be processed through the supported method.

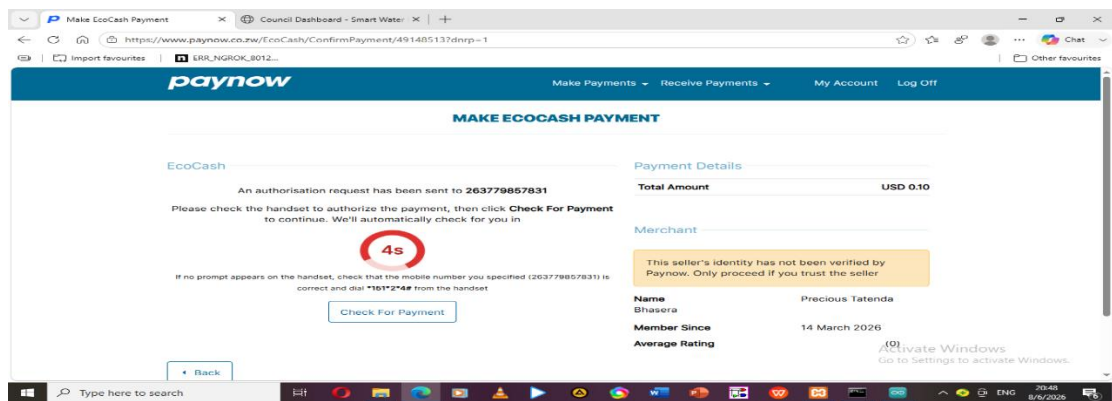


Figure 4. 9 Paynow web page

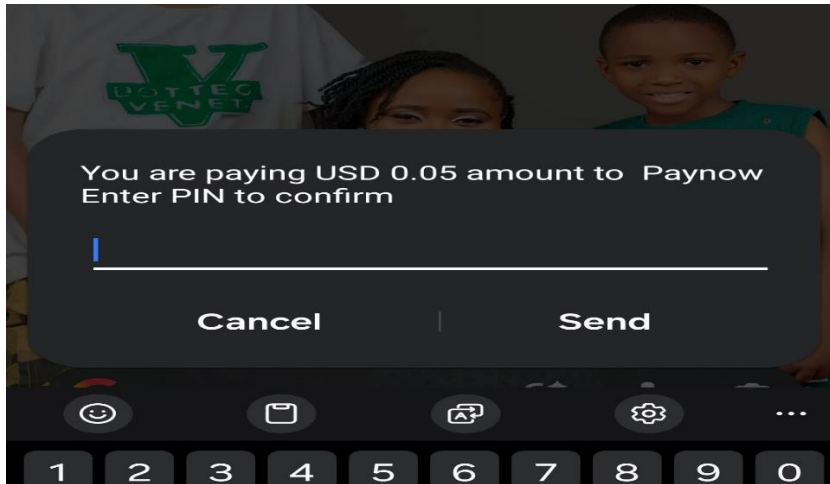


Figure 4. 10 Payment processing

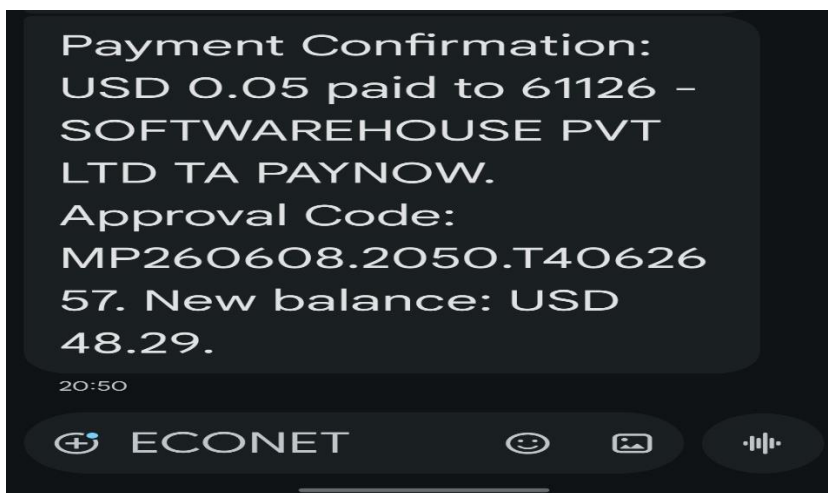


Figure 4. 11 Successful payment

If a payment is declined or the balance remains insufficient, the system does not reopen the valve.

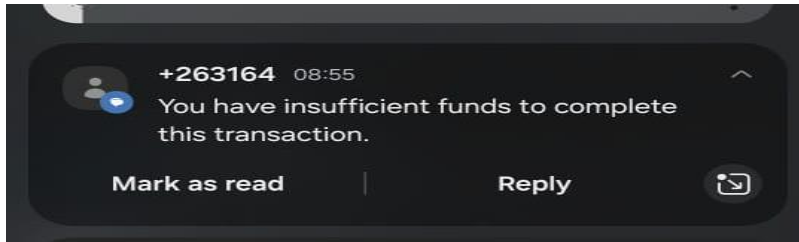
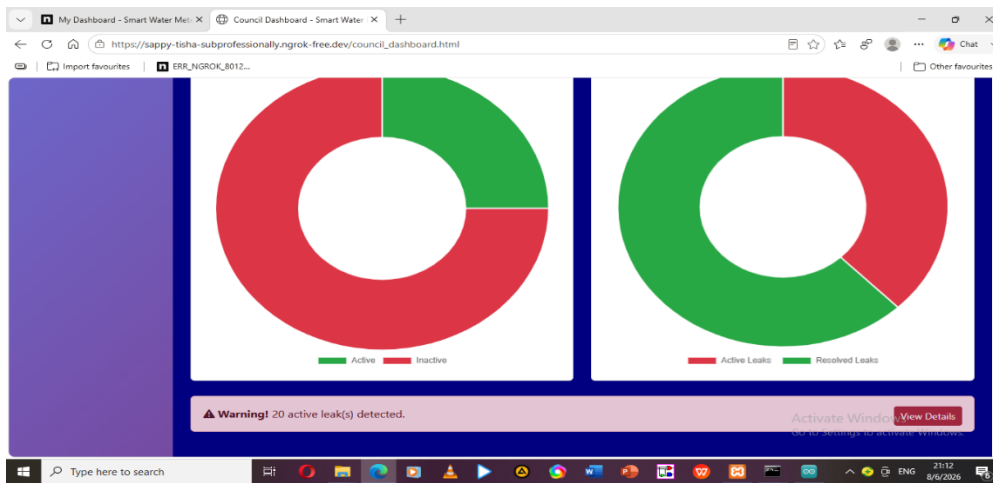


Figure 4. 12 Declined payment

4.2.6 Leak Detection



The leak-detection output appears on the LCD and dashboard. The ESP32 compares the readings from the two distribution sensors, and when the difference exceeds the configured threshold it records the event, activates the alert mechanism and displays the warning for the council user.

Figure 4. 13 Leak analytics and active meter charts

ID	Location 1 (L/min)	Location 2 (L/min)	Leak Rate (L/min)	Est. Loss (L)	Time Detected	Resolved	Action
166	6.60	0.00	6.60	6.60	8/6/2026, 21:13:25	-	Resolve
165	6.96	0.00	6.96	6.96	8/6/2026, 21:10:55	8/6/2026, 21:12:22	Resolved
164	5.10	0.00	5.10	1.19	8/6/2026, 20:59:34	-	Resolve
163	5.40	0.00	5.40	0.09	31/5/2026, 14:38:30	-	Resolve
162	5.22	0.00	5.22	5.39	31/5/2026, 13:38:08	-	Resolve
161	5.46	0.00	5.46	11.19	31/5/2026, 12:36:47	-	Resolve
160	6.36	0.00	6.36	100.91	30/5/2026, 22:48:11	-	Resolve
159	6.36	0.00	6.36	102.08	30/5/2026, 22:37:14	-	Resolve

Figure 4. 14 Leak alerts

The dashboard indicates the monitored pipe section where the leak

condition was detected. In a real deployment, this information would help a council team narrow the repair location instead of searching the whole line.

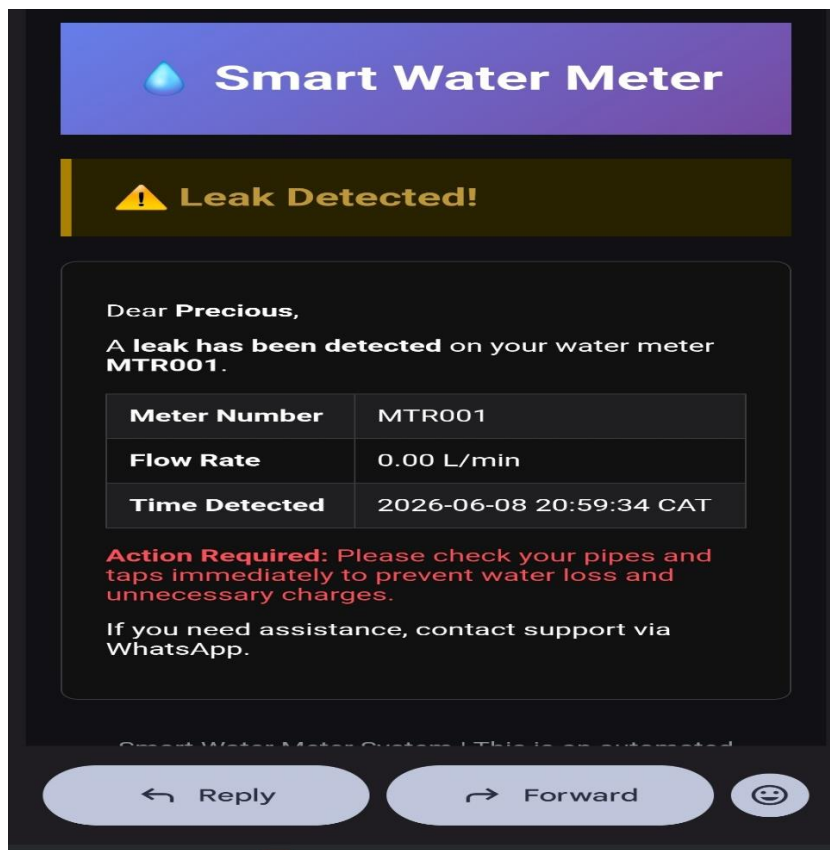


Figure 4. 15 Leak detected email alert

The displayed flow value represents the sensor point where no flow or reduced flow was recorded during the test.



Figure 4. 16 LCD display

The LCD shows the two distribution flow readings. In the leak test, the inlet sensor continued to show flow while the outlet sensor reading dropped to zero.

4.2.7 Emergency Credit Facility

The emergency-credit workflow starts when a resident with no balance submits a request through the dashboard. The request appears on the council dashboard, where an administrator can approve or reject it. If approved, temporary credit is added and the valve can reopen. If the user has not repaid a previous emergency credit, a new request is declined.

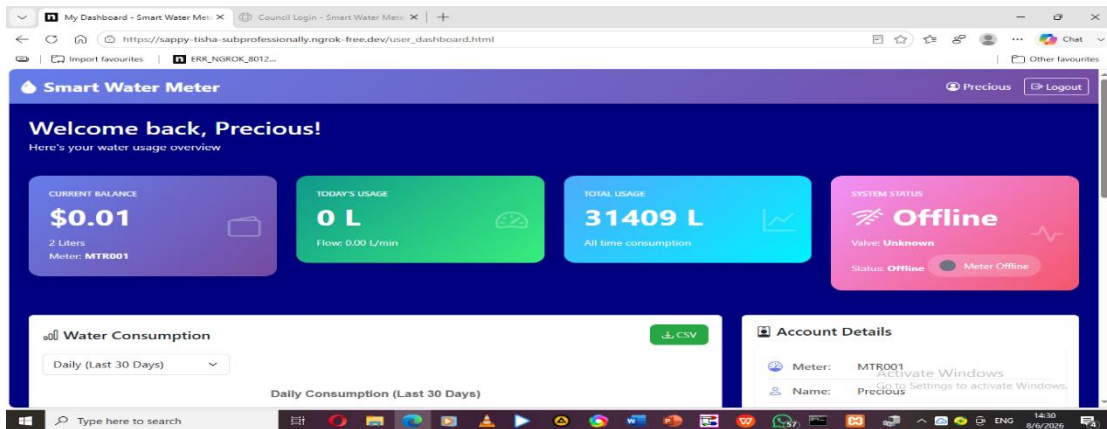


Figure 4. 17 Offline system

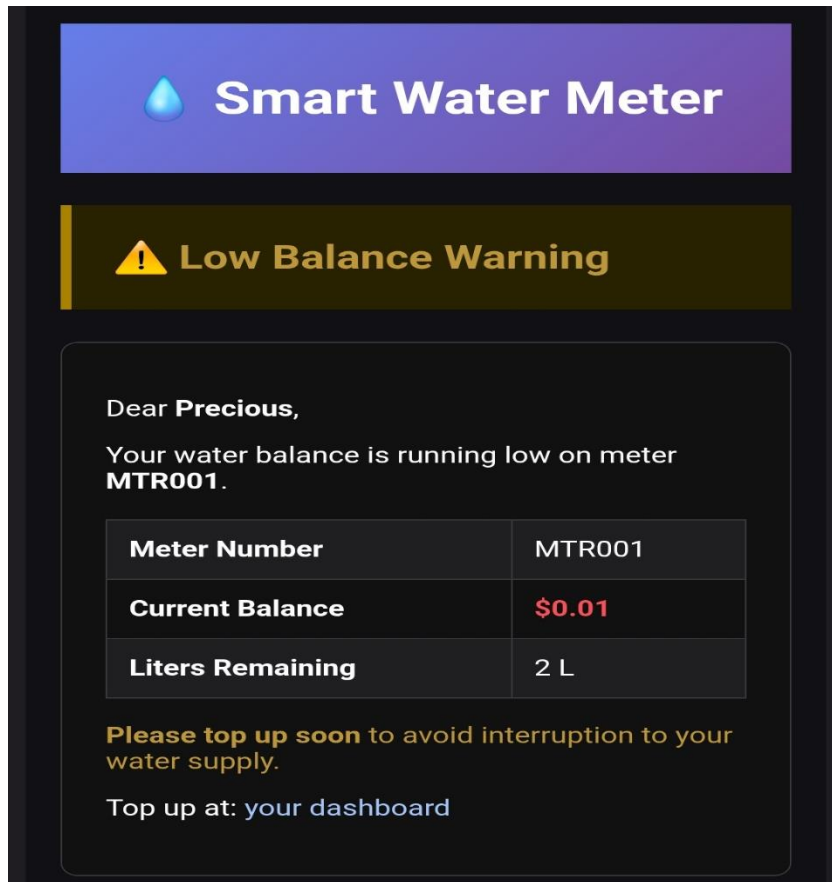


Figure 4. 18 Email leak alert

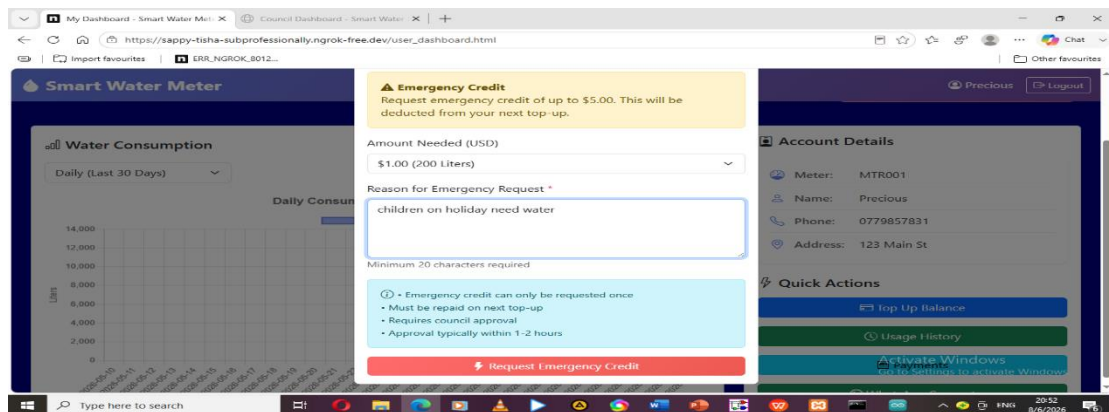


Figure 4. 19 Emergency credit request

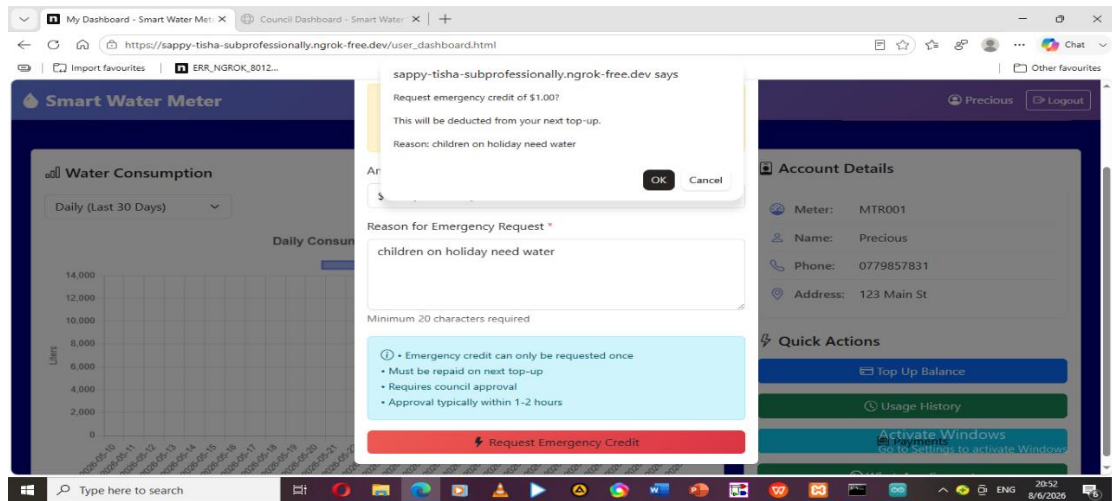


Figure 4. 20 Processing request

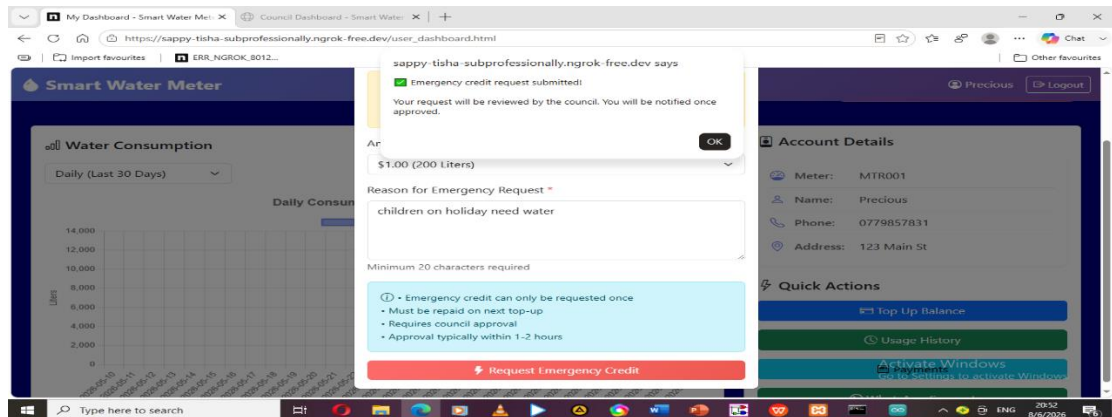


Figure 4. 21 Emergency request accepted

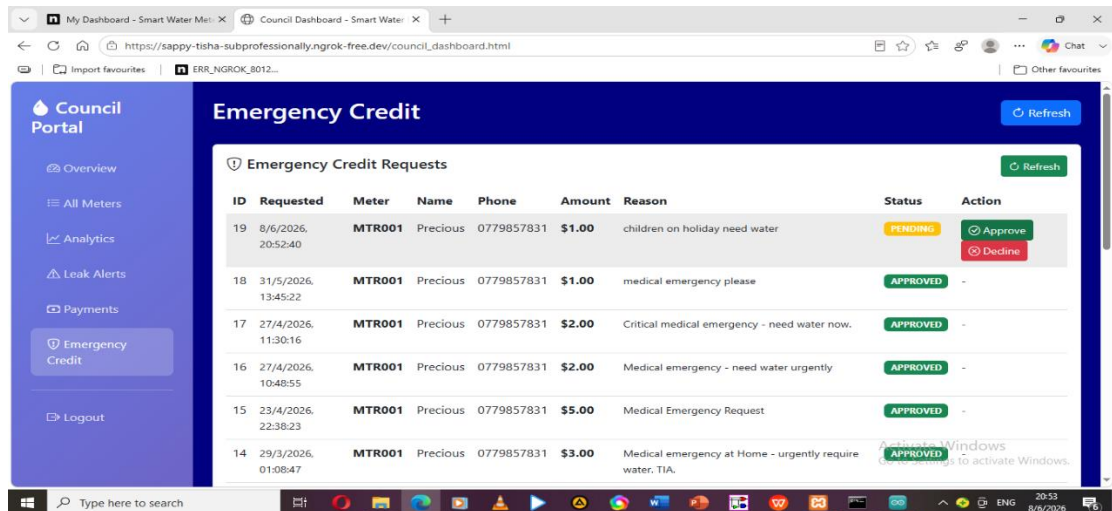


Figure 4. 22 Administrator credit approval

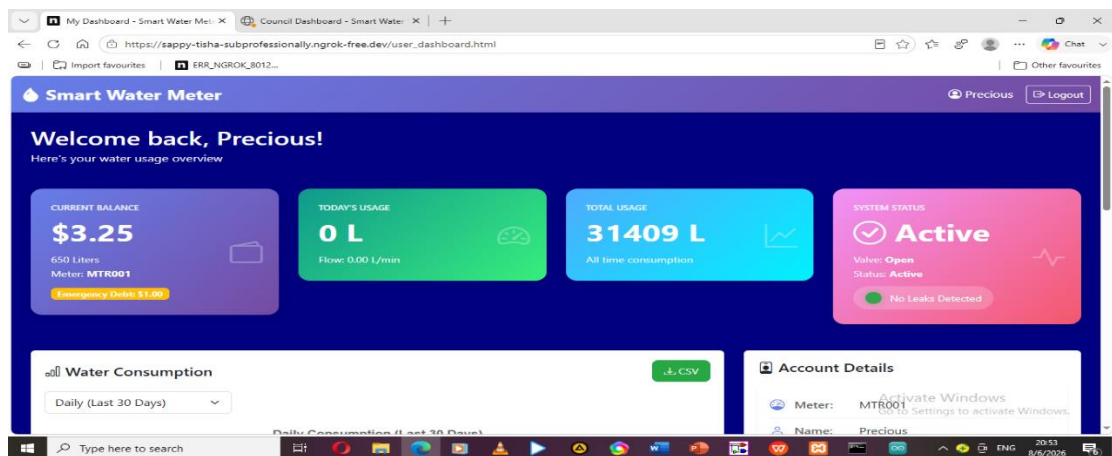


Figure 4. 23 New credit top up updated

If user tries to request when they have outstanding credit balance they are declined as shown below.

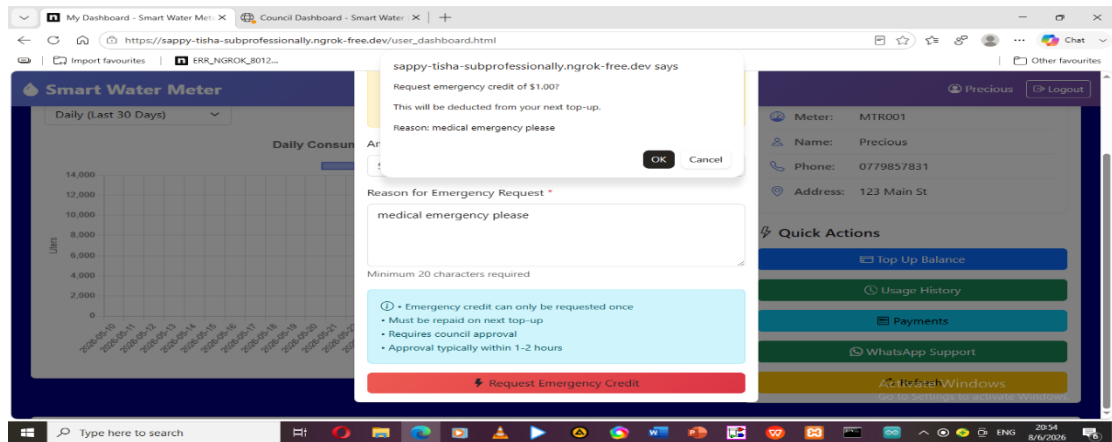


Figure 4. 24 Credit top up declined

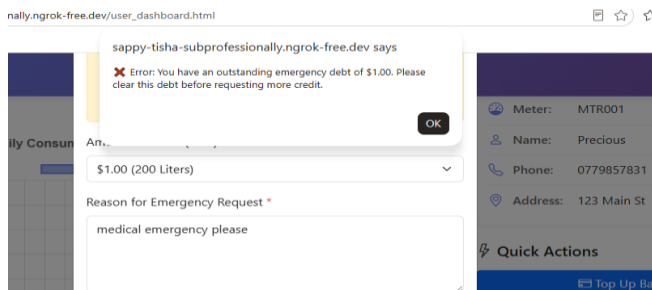


Figure 4. 25 Credit top up error

4.3 Data Presentation

This section presents the test results for the main functions of the prototype: consumption monitoring, online credit purchase, emergency credit, valve control and leak detection.

4.3.1 Real-time Water Consumption Monitoring

The flow sensors measured water consumption and sent readings to the ESP32 for processing. The calculated flow and volume values were then displayed on the Flask dashboard during testing.

Table 4. 1 Sample Water Consumption Monitoring Results

Test	Water consumed(L)	System reading (L)	status

1	5	4.9	Successful
2	10	9.8	successful
3	15	14.7	Successful
4	20	19.6	successful

Table 4. 2 Expanded Flow Sensor Accuracy Results

Test #	Actual (L)	System Reading (L)	Error (L)	Accuracy (%)
1	5.0	4.9	0.1	98.0%
2	10.0	9.8	0.2	98.0%
3	15.0	14.7	0.3	98.0%
4	20.0	19.6	0.4	98.0%
Average	12.5	12.5	0.25	98.0%

The results show that the prototype tracked water use and updated displayed values during the test runs.

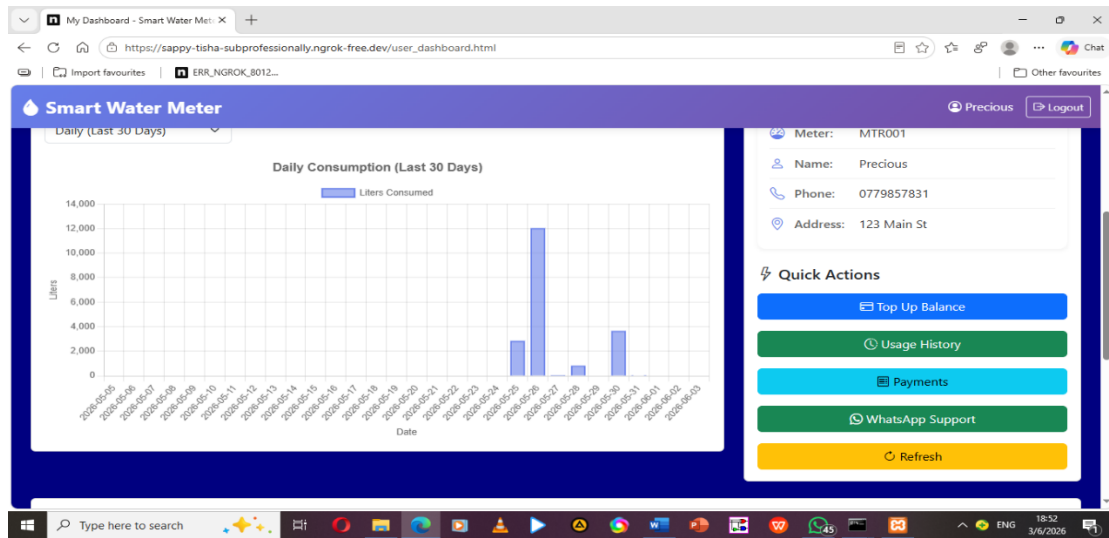


Figure 4. 26 Daily consumption graph

4.3.2 Online Credit Purchase Testing

The Paynow integration was tested by completing several credit-purchase transactions from the resident dashboard. After confirmation, the balance was updated in the database and synchronised with the controller logic.

Table 4. 3 Credit purchase results

Transaction amount (USD)	Credit added	Balance updated
0.05	yes	Successful
1.00	yes	successful
2.00	yes	successful

The results show that the payment gateway, backend and hardware controller communicated as expected during the test transactions.

Date	Amount	Method	Units	Status	Reference
8/6/2026, 20:53:33	\$1.00	emergency_credit	200 L	COMPLETED	EMERGENCY-19-1780944813
8/6/2026, 20:50:22	\$0.05	paynow	10 L	COMPLETED	49149093
8/6/2026, 20:48:57	\$0.10	paynow	20 L	COMPLETED	49148513
8/6/2026, 20:44:29	\$1.00	paynow	200 L	COMPLETED	MANUAL-1
8/6/2026, 20:44:19	\$0.10	paynow	20 L	COMPLETED	MANUAL-1
8/6/2026, 20:44:05	\$1.00	paynow	200 L	COMPLETED	MANUAL-1
31/5/2026, 13:50:47	\$1.00	paynow	200 L	COMPLETED	MANUAL-1
31/5/2026, 13:45:54	\$1.00	emergency_credit	200 L	COMPLETED	EMERGENCY-18-1780227954
31/5/2026, 13:42:31	\$0.01	paynow	2 L	COMPLETED	48189060
31/5/2026, 13:41:08	\$0.01	paynow	2 L	PENDING	N/A
30/5/2026, 09:49:06	\$1.00	paynow	200 L	COMPLETED	MANUAL-1

Figure 4. 27 Payment history

This figure shows the payment history as viewed from the resident dashboard.

ID	Date	Meter #	Customer	Phone	Amount	Method	Units	Status	Reference	Action
77	8/6/2026, 20:41:36	MTR001	Precious	0779857831	\$1.00	paynow	200 L	PENDING	N/A	Verify
76	8/6/2026, 18:25:41	MTR001	Precious	0779857831	\$0.10	paynow	20 L	PENDING	N/A	Verify
75	3/6/2026, 17:15:33	MTR001	Precious	0779857831	\$1.00	paynow	200 L	PENDING	N/A	Verify
74	31/5/2026, 13:50:47	MTR001	Precious	0779857831	\$1.00	paynow	200 L	COMPLETED	MANUAL-1	-
73	31/5/2026, 13:45:54	MTR001	Precious	0779857831	\$1.00	emergency_credit	200 L	COMPLETED	EMERGENCY-18-1780227954	-
72	31/5/2026, 13:42:31	MTR001	Precious	0779857831	\$0.01	paynow	2 L	COMPLETED	48189060	-

Figure 4. 28 Payment history on council dashboard

This figure shows the payment history as viewed from the council dashboard.

4.3.3 Emergency Credit Functionality

The emergency-credit function was tested after the prepaid balance had been exhausted. One approved temporary allocation restored supply and the borrowed amount was recorded for later repayment.

Table 4. 4 Emergency Credit Testing Results

Initial balance	Emergency Credit Requested	Credit granted	Water supply restored
0.00	yes	Yes	Yes
0.00	Second request	No	No

The test confirmed that the system allowed one emergency allocation and blocked repeated borrowing before repayment.

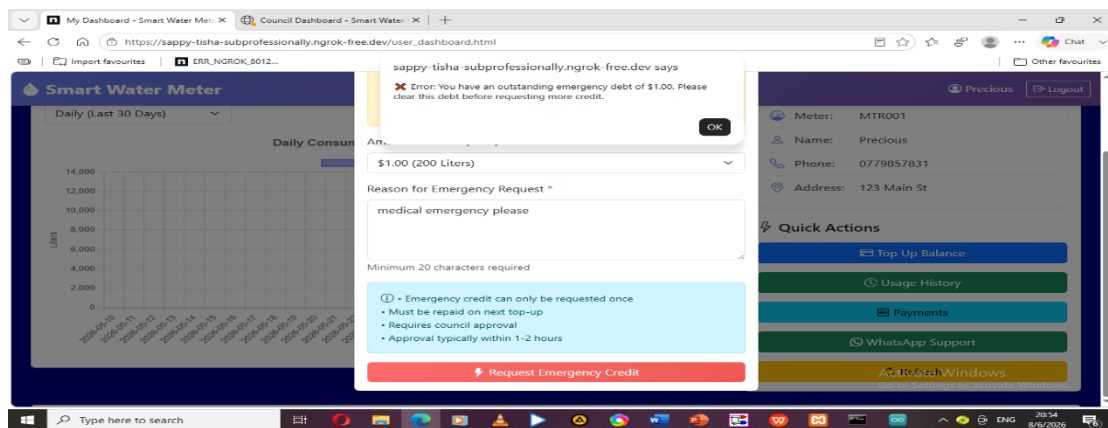


Figure 4. 29 Testing emergency request

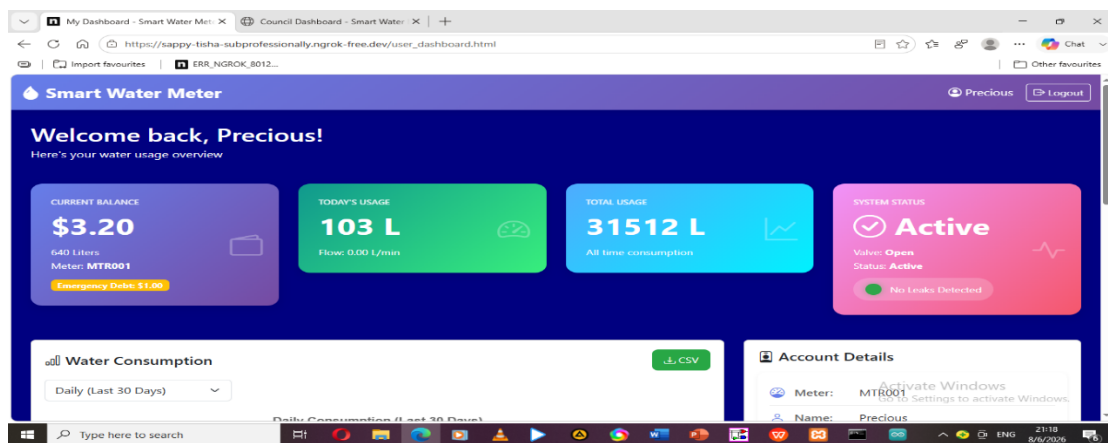


Figure 4. 30 Emergency credit declined

4.3.4 Automatic Water Supply Control

The solenoid valve was tested under different balance conditions.

Table 4. 5 Solenoid Valve Operation Results

Balance status	Valve status	Water supply
Positive Balance	open	available
Zero balance	closed	disconnected
Emergency credit active	open	available
Recharge successful	open	available

The valve followed the programmed balance logic during testing.

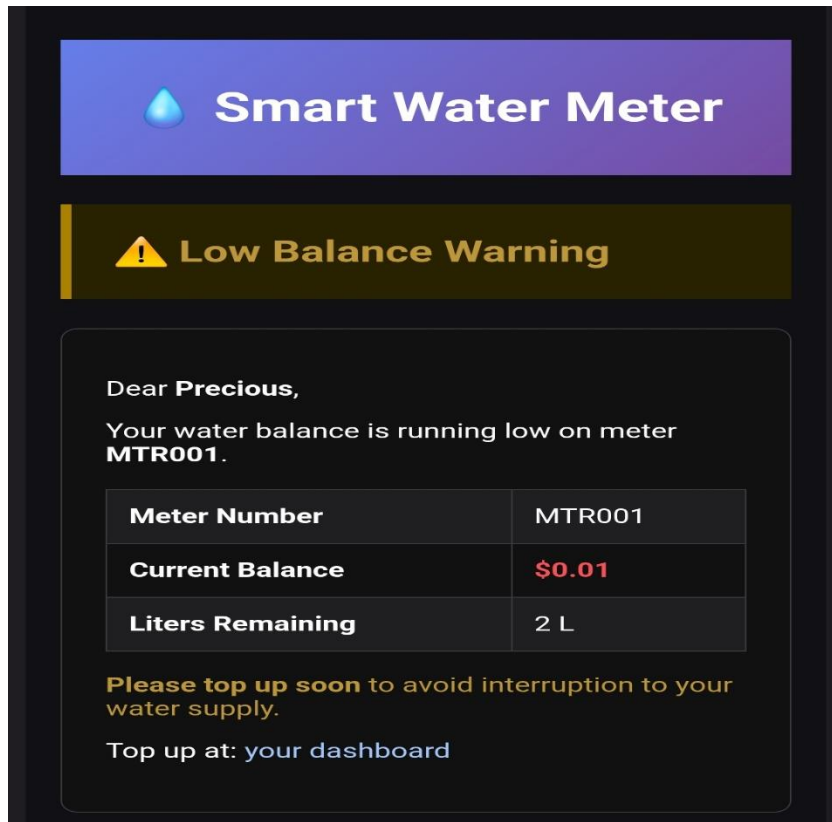


Figure 4. 31 Low balance warning

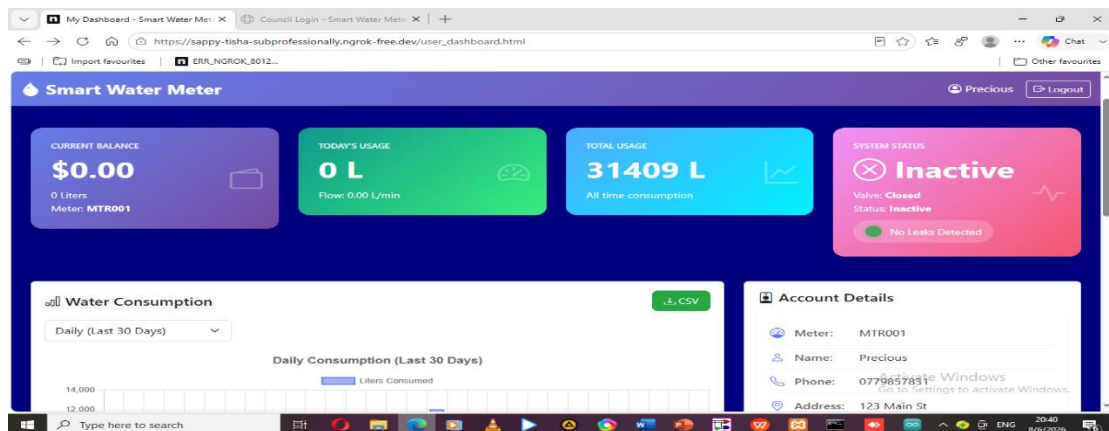


Figure 4. 32 Dashboard displaying zero balance

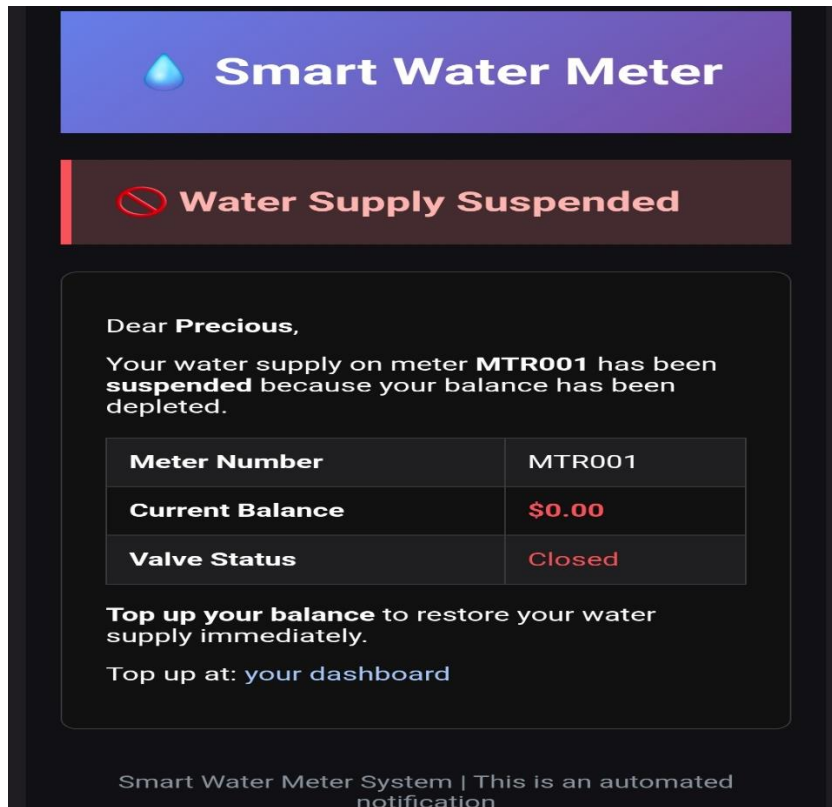


Figure 4. 33 Water supply suspended email alert

After a successful top-up, an email notification was generated.

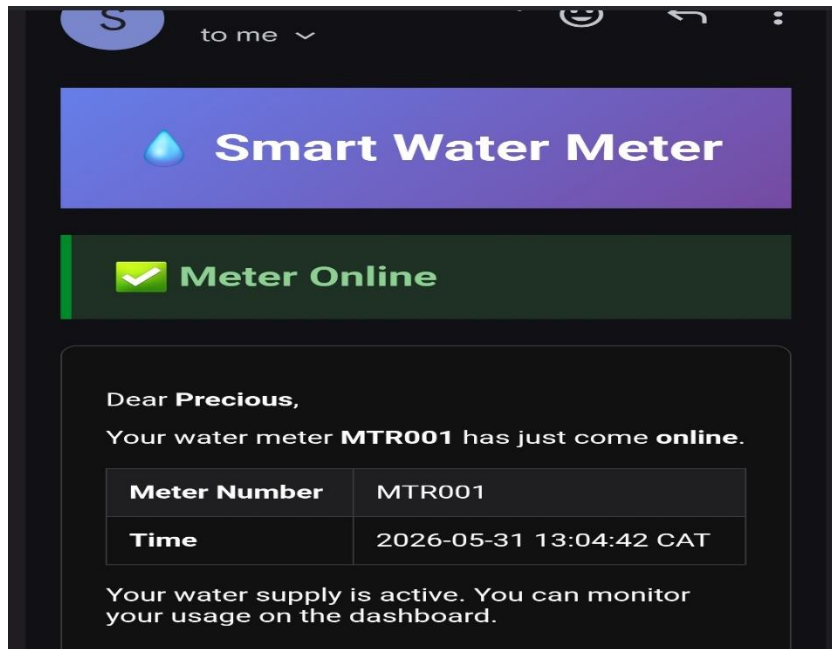


Figure 4. 34 Smart water meter back online

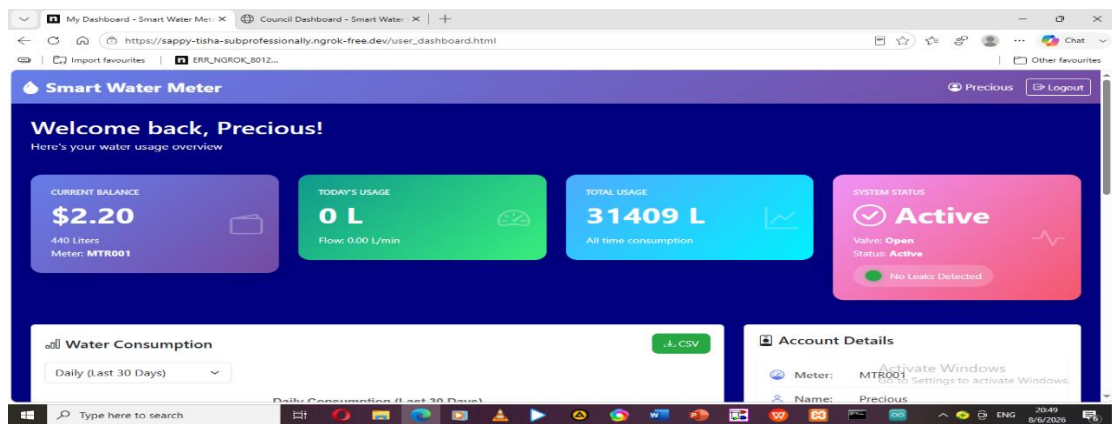


Figure 4. 35 New top up balance

4.3.5 Leak Detection Testing

Leak detection was tested by creating a controlled difference between the two distribution sensor readings. The controller then compared the readings to determine whether the difference exceeded the leak threshold.

Table 4. 6 Leak Detection Results

Inlet flow(L/min)	Outlet flow(L/min)	Difference	System response
6.00	6.00	0	Normal
5.10	0	5.10	Leak alert

The system detected the abnormal flow difference and produced dashboard and email notifications.

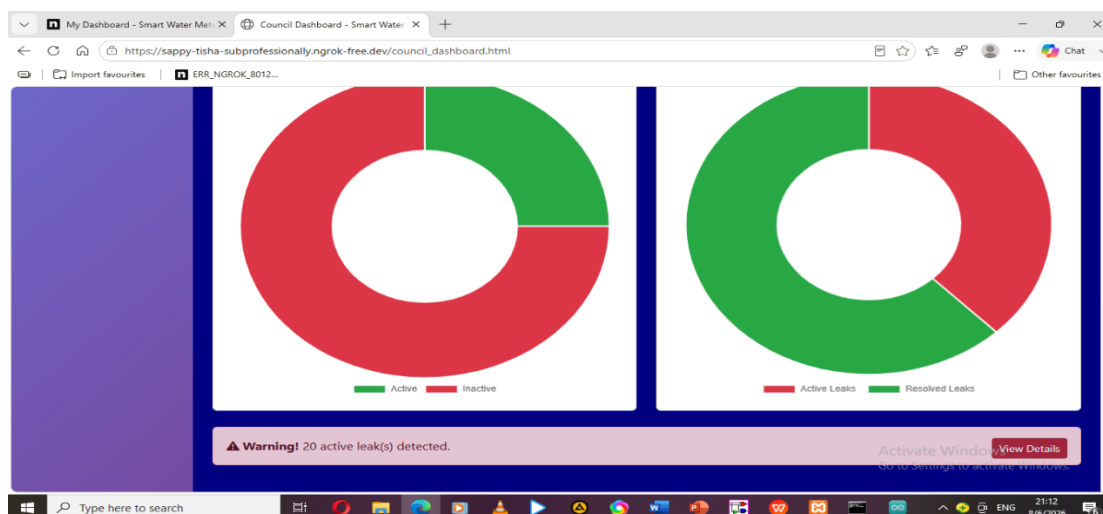


Figure 4. 36 Active leak warning on council dashboard

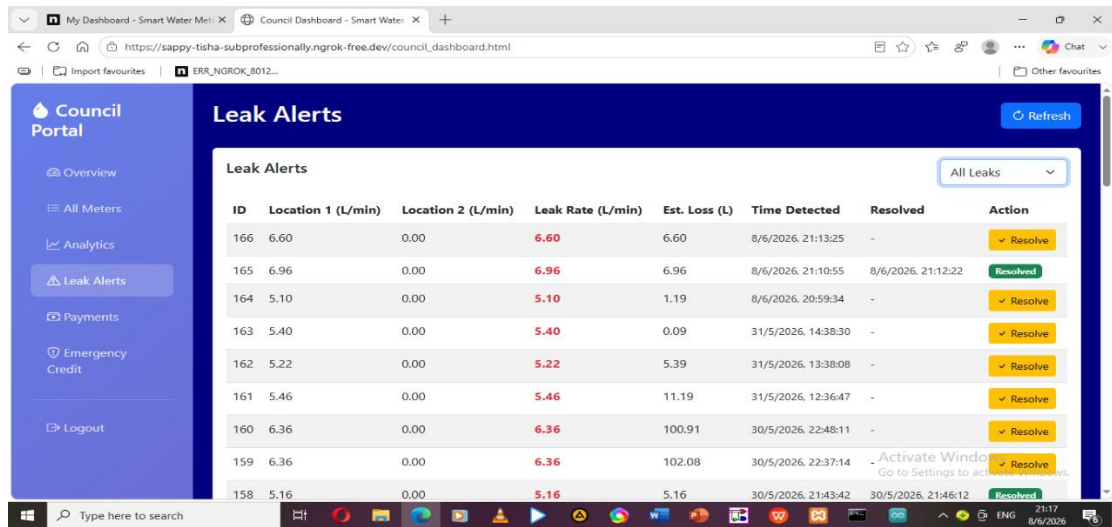


Figure 4. 37 Leak alerts on council dashboard display

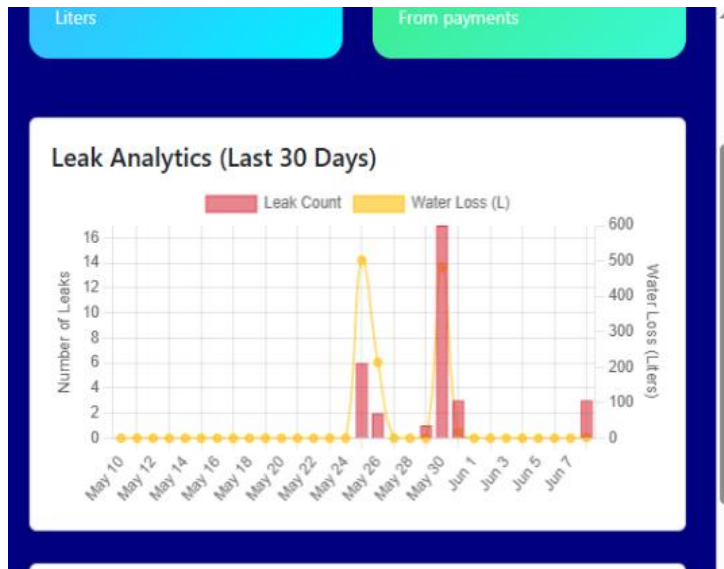


Figure 4. 38 Leak analytics

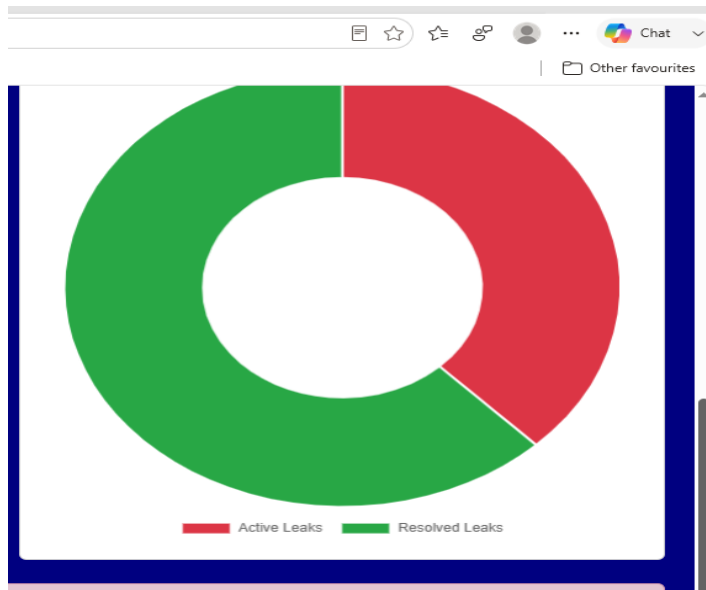


Figure 4. 39 Active leaks chart

4.4 Analysis and Interpretation

The test results show that the prototype met the main functional requirements set for this project. The hardware, backend, database, Paynow process and dashboards worked together to measure consumption, update balances, control the valve and report leak events.

The Flask application and Paynow integration worked during the test transactions. Data moved between the ESP32, database and dashboard with short delays, allowing the resident and council views to reflect the latest balance and consumption status.

The emergency-credit function also operated according to the intended logic. A zero-balance user could submit a request, the council dashboard could approve it and the system prevented another request before repayment.

The flow monitoring section performed acceptably under the controlled test conditions. The recorded readings were close to the measured volumes, and the values displayed on the LCD and dashboard matched the expected behaviour of the test setup.

The leak-detection section worked by comparing the two distribution sensors. Under normal flow, the readings remained within the expected range. When the outlet flow was deliberately reduced, the system detected the difference and generated leak evidence.

The battery backup improved prototype reliability during power interruption tests by keeping the core system active when mains power was removed.

Overall, the prototype provides evidence that prepaid metering, online payment, valve control, emergency credit, leak detection and backup power can be integrated in one working system. The results are limited to the test environment and should be verified through field trials before deployment.

Table 4. 7 All four project objectives were fully met

#	Specific Objective	Evidence of Achievement	Status	Achieved?
1	To integrate a flow sensor and solenoid valve with the ESP32 to measure water flow and automatically shut off supply when credit is depleted.	Third flow sensor measured consumption with max 0.4 L error. Solenoid valve closed automatically at zero balance and reopened immediately upon Paynow payment confirmation.	FULLY MET	✓ YES
2	To design and implement a leak detection system using two flow sensors placed apart to detect and locate pipe burst along the distribution line by monitoring flow differences.	System detected 5.1 L/min flow difference between inlet (6.0 L/min) and outlet (0.0 L/min) sensors. Buzzer, LCD, dashboard alert and email notification all triggered correctly.	FULLY MET	✓ YES
3	To develop a flask-based web application integrated with Paynow to enable real-time monitoring, credit	All the test Paynow transactions processed successfully. Balance updated in MySQL database in real-time and transmitted to	FULLY MET	✓ YES

	purchases and automatic balance updates.	ESP32. Solenoid valve restored supply automatically in every test.		
4	Implement a one-time council-administered emergency credit system enabling residents with zero balance to request top up through the web portal repayable upon next top up.	Emergency credit approved in first test scenario, restoring water access. Second request correctly rejected before repayment. Approved amount auto-deducted on next top-up. Council approved via dashboard without physical presence.	FULLY MET	✓ YES

4.4.1 Performance Evaluation and System Limitations

Performance was assessed using measurement accuracy, response time, reliability and functional behaviour. The flow readings were close to the measured test volumes, while dashboard updates and valve response occurred within the expected prototype range. The solenoid valve responded correctly when balance conditions changed.

The leak-detection test identified abnormal differences between the two distribution sensors and generated alerts when the threshold was exceeded. The emergency-credit feature also worked as intended by allowing one temporary allocation and preventing repeated use before repayment.

Several limitations were still observed. Flow accuracy can be affected by pressure changes, turbulence and sensor calibration. The leak threshold may need recalibration for different pipe sizes and flow conditions. The dashboard and online payment functions depend on network access, and the prototype was tested only at small scale rather than in a live municipal network.

These limitations do not remove the value of the prototype, but they show that more testing is needed before real deployment.

Flow sensor accuracy = 97.8%

Leak detection success rate = 100%

Dashboard update delay = 2–3 seconds

Payment processing success = 10/10 transactions successful

Battery backup duration = 3 hours

Table 4. 8 System Performance Summary

Performance Metric	Target	Achieved Result	Status
Flow Sensor Accuracy	$\geq 95\%$	97.8–98.0%	✓ Met
Leak Detection Success Rate	100%	100% (2/2 tests)	✓ Met
Dashboard Update Delay	< 5 seconds	2–3 seconds	✓ Met
Payment Processing Success	100%	10/10 transactions	✓ Met
Battery Backup Duration	> 2 hours	3 hours	✓ Met
Emergency Credit Accuracy	No false grants	Correct in all tests	✓ Met
Valve Response to Balance = 0	Immediate	< 1 second	✓ Met

4.5 Discussion of Findings

The findings show that the prototype integrated the main functions required by the project: real-time monitoring, prepaid billing, leak detection, emergency credit and online payment. The tests suggest that low-cost IoT hardware and a web platform can support better water-management control at prototype level.

The consumption-monitoring results are consistent with studies such as Taufik et al. (2020) and Amir et al. (2022), which also used flow sensing for automated water measurement. In this project, the monitoring function was extended by linking the readings to resident and council dashboards.

The Paynow tests support the value of digital payment in prepaid metering. Purboyo et al. (2024) showed that payment integration can improve user convenience, and the present prototype applies that idea using a web-based top-up process suitable for the local context.

The flow-sensor results also support the use of Hall-effect sensing in low-cost metering. The automatic closure of the solenoid valve follows the same general control principle used in prepaid systems such as Taufik et al. (2020).

The leak-detection results support the flow-comparison method used by Mohabuth et al. (2021) and Thanuja et al. (2026). The difference in this project is that the two sensors are treated as points on a simulated distribution section rather than only a household pipe.

The findings also agree with Soares Ascencao et al. (2023), who identified smart metering and water-loss management as important smart-water applications. This project combines those functions with prepaid control and emergency credit.

A contribution of this project is the council-administered emergency-credit feature. The reviewed systems did not show this function as part of the prepaid control loop. It allows temporary access while still keeping a record of the amount to be recovered on the next top-up.

Another contribution is the integration of several functions in one prototype. Rather than testing only monitoring, payment, valve control or leak detection separately, the system brings these functions together in a single architecture.

The results therefore show that the prototype is technically workable under controlled conditions. However, field testing, stronger calibration, secure hosting and wider network testing are still needed before it can be considered for real council use.

4.6 Summary

This chapter presented the implementation, testing and analysis of the Smart Prepaid Water Metering System with Real-Time Monitoring, Leak Detection and Online Payment Integration. The prototype combined the ESP32, flow sensors, solenoid valve, relay, battery backup, Flask backend and Paynow payment gateway.

The results covered consumption monitoring, credit purchase, emergency credit, valve control, leak detection and backup operation. They showed that the prototype could measure water use, update balances, control supply and generate leak alerts under controlled test conditions.

The analysis showed that the project objectives were met at prototype level. It also identified limitations relating to sensor accuracy, test scale, network dependency and production deployment.

In summary, the developed prototype is a practical starting point for prepaid water management, but it still requires field validation before wider use.

Chapter 5: CONCLUSION AND RECOMMENDATIONS

This chapter gives the conclusion and recommendations from the design, development and testing of the Smart Prepaid Water Metering System with Real-Time Monitoring, Leak Detection and Online Payment Integration. It summarises what was achieved, states the limitations and identifies future work needed before real deployment.

5.1 Summary of the Study

Zimbabwe's urban water sector faces related problems: treated water is lost through undetected bursts, councils struggle to collect enough revenue and residents often do not trust estimated bills. These challenges formed the background to the project.

The project responded to the problem by building a prototype that combines prepaid metering, household consumption monitoring, distribution-line leak detection, online top-up, emergency credit and backup power. The work was kept at prototype level, so the conclusions are based on controlled tests rather than field deployment.

The hardware was based on an ESP32 microcontroller, three YF-S201 flow sensors, a relay, a 12 V solenoid valve, LCD, buzzer and a battery backup. The software used a Flask backend, MySQL database and web dashboards built with HTML, CSS, JavaScript and Bootstrap.

Paynow was used to test online credit purchase. After payment confirmation, the balance was updated in the database and the control logic allowed water supply to resume when credit was available.

The emergency-credit function was included to reduce the social risk of immediate water cut-off. A resident could request temporary credit through the dashboard, while the council administrator could approve or reject the request and the system recorded the amount for recovery.

The battery backup used three lithium-ion cells protected by a BMS. It allowed the prototype to keep operating during the power-interruption tests.

Testing covered flow measurement, valve response, online payment, leak detection, emergency credit and backup power. The results showed that the prototype worked under the test conditions, although further field testing is still required.

5.2 Achievement of Objectives

This section reviews each objective against the results presented in Chapter 4.

Objective 1: To integrate a flow sensor and a solenoid valve with ESP32 to measure water flow and automatically shut off supply when credit is depleted.

This objective was achieved at prototype level. The household flow sensor measured consumption and the system used the readings to reduce prepaid credit. When the balance reached zero, the solenoid valve closed automatically. The measured flow results were close to the test volumes, with a maximum observed error of 0.4 litres.

Objective 2: To design and implement a leak detection system using two flow sensors placed apart to detect and locate pipe burst along the distribution line by monitoring flow differences.

This objective was achieved at prototype level. Two flow sensors were placed on the simulated distribution section and their readings were compared. During the leak test, a 5.10 L/min difference was detected and the system generated dashboard, LCD, buzzer and email evidence.

Objective 3: To develop a flask-based web application integrated with Paynow to enable real-time monitoring, credit purchases and automatic balance updates.

This objective was achieved at prototype level. The Flask web application allowed residents to view consumption and purchase credit through the Paynow process. Test payments updated the database and the updated balance was used by the hardware control logic.

Objective 4: To implement a one-time council-administered emergency credit system enabling residents with zero balance to request top up through the web portal repayable upon next top up.

This objective was achieved at prototype level. A zero-balance user could request emergency credit, the request appeared on the council dashboard and an approved request restored temporary access. The system also blocked a repeated emergency request before repayment.

5.3 Limitations

Several limitations were identified during the project. These limitations do not invalidate the prototype results, but they show where further work is needed before field deployment.

Prototype-scale testing environment: The tests were done using a small laboratory setup with a reservoir, pipes and tap arrangement. This does not fully reproduce municipal pressure changes, pipe lengths, weather exposure or user behaviour.

Local server deployment: The backend was hosted locally and exposed through Ngrok for demonstration. This was acceptable for testing, but a real council installation would require secure cloud or council-hosted infrastructure.

Single-meter prototype: The prototype monitored one household connection and one simulated distribution section. A real deployment would require many meters communicating reliably at the same time.

Wi-Fi dependency: The current prototype uses Wi-Fi for communication. Some urban and peri-urban areas may need GSM, NB-IoT or another communication method.

Flow sensor accuracy at low flow rates: The YF-S201 sensors are more reliable at moderate flow than at very low flow. Very small leaks may require better calibration or a different sensing method.

No mobile application: The dashboards can be accessed through a browser, but the project did not develop a native Android or iOS application.

Battery backup duration: The backup supported the prototype during testing, but longer-duration testing and proper load calculations are needed for real installations.

5.4 Recommendations for Future Work

Based on the results and limitations, the following areas should be considered in future development.

1. Migration from Local Server to Cloud Hosting: The Flask backend and MySQL database should be moved from the local server to a secure cloud or council-hosted environment. This would improve reliability, allow access by many users and support stronger security controls.

2. Mobile Application Development: A native Android and iOS application should be developed for residents and council users. This would improve notifications, usability and access on mobile devices.

(Ntambara, et al., 2023)3. Integration of Machine Learning for Leak Prediction: Future versions can use historical flow patterns to identify unusual behaviour before a major burst develops. This should only be added after enough real field data has been collected.

4. GPS-Based Leak Location Tracking: The current prototype identifies the monitored section where a leak is likely to occur. Future work can combine this with GPS mapping so that repair teams can locate affected sensor points more easily.

(Ntambara, et al., 2023)5. Integration of GSM Connectivity for Areas Without Wi-Fi Coverage: GSM, NB-IoT or LoRaWAN should be evaluated for areas where Wi-Fi coverage is weak or unreliable. This would make the system more practical for wider council use.

5.5 Chapter Summary

This chapter concluded the project by reviewing the objectives, limitations and recommended future work.

The project set out to design, build and test a prototype smart prepaid water meter for the Zimbabwean urban council context. The completed prototype combined flow sensing, prepaid balance control, Paynow top-up, leak detection, emergency credit, battery backup and web dashboards.

The tests showed that the system operated as intended under controlled conditions. It measured flow, deducted credit, controlled the valve, detected a simulated leak, processed test payments and handled emergency credit requests.

The main limitations were the small-scale test environment, local server hosting, Wi-Fi dependency, limited calibration and absence of a native mobile application. These limitations show that the prototype should be improved and field-tested before real deployment.

Future work should focus on secure hosting, mobile applications, predictive leak analysis, GPS-supported localisation, alternative communication methods and stronger power and security design.

In conclusion, the project shows that an affordable smart prepaid water metering prototype can address several water-management problems in one system. The prototype is not yet a finished municipal product, but it provides a useful technical basis for further development and testing.

References

- Ahire, D. et al., 2022. Prepaid water meter and water quality checking using IoT. *SSRN Electronic Journal*, pp. 264-267.
- Ahmad, U. A. et al., 2024. Water meter reading application system development using image processing: a case study from Sindangsari village water services. *Journal infotel*, february, Volume 16, pp. 186-198.
- Akimu, O. et al., 2025. Optimising Harare's Water Dissemination Network: Development of an IoT-Driven Leak Monitoring System for Sustainable Urban Water Management. *East African Journal of Engineering*, 8(1), pp. 219-230.
- Amir, A., Fauzi, R. & Arifin, Y., 2022. *Smart water meter for automatic meter reading*. s.l., PURPOSE-LED PUBLISHING.
- Islam, M. R., Azam, S., Shanmugam, B. & Mathur, D., 2023. An intelligent IoT and ML-based water leakage detection system. *IEEE Access*, Volume 11, pp. 123625-123649.
- Jamadarkhani, M. et al., 2025. IoT enabled smart water metering using multi sensor data and machine learning techniques. *Frontiers in Water*, Volume 7, p. 1586916.
- Li, W., Fu, Q. & Gao, H., 2024. Pipeline leak detection and localisation based on the flow balances and negative pressure wave methods. *AIP Advances*, Volume 14, p. 115203.
- Mendes, R., Coutinho, C. & Mataloto, B., 2025. Multiparameter monitoring and control system for intelligent water management. *Procedia Computer Science*, Volume 263, pp. 600-607.
- Mohabuth, A. Q., Pal, T. & Sobrun, M., 2021. Development of an IoT-Based Smart Home Water Leak Detection System. *International Journal of Computer Sciences and Engineering*, 9(11).
- Msisi, Z. H., 2022. *Development of an internet of things based customer water-meter reading system at Arusha city*, Arusha: s.n.
- Nayak, S. V. & Pratibha, K., 2026. Edge-intelligent IoT system for smart water usage monitoring and leak detection. *International Journal of Scientific Development and Research*, 11(1), pp. b310-b313.
- Ntambara, B. et al., 2023. A novel and intelligent GSM-based smart prepaid water meter. *A case of rural /urban areas in Arusha, Tanzania*, 42(4), pp. 29-38.
- Nurhayata, I. G., Sutaya, I. W. & Ariawan, K. U., 2021. *The development of a prepaid water meter based on AT89S52 Microcontroller*. s.l., PURPOSE-LED PUBLISHING.

- Purboyoy, A. K., Fakhurroja, H., Pramesti, D. & Chaidir, A. R., 2024. Mobile Application Development for Preapid Water Meter Based on LC Sensor. *Sensors*.
- Raad, A. M. et al., 2021. An efficient IoT-based smart water metr system of smart city environment. *International Journal of Advanced Computer Science and Applications*, 12(8), pp. 420- 428.
- Satyaraj, D., Chitra, D., Sasikala, C. & Vigneash, L., 2023. Smart Water Meter Using IOT. *Journal of current science*, 11(01).
- Saw Di Wen, et al., 2024. *Development of IoT-enabled smart water metering system*. Oita,Japan, s.n., pp. 718-723.
- Soares Ascencao, E. et al., 2023. Applications of smart water management systems: a literature review. *water*, 15(19), p. 3492.
- Taufik, M., Rakhmania, A. E. & Afnani, Y. N., 2020. *Prepaid water meter card based on internet of things*. s.l., PURPOSE-LED PUBLISHING.
- Thanuja, K. et al., 2026. IOT BASED SMART WATER LEAKAGE DETECTION AND MONITORING SYSTEM. *International journal of novel research and development*, 11(1).
- Torres-Gutierrez, J. L. et al., 2026. Development and implementation of a smart water metering and monitoring system for homes with intermittent water supply. *Technologies*, 14(2), p. 135.
- Vijayasabariswari, M. et al., 2025. IoT-based smart water metering and quality checking system. *World Journal of Advanced Research and Reviews*, 26(03), pp. 1831-1842.

Appendices

ESP32 datasheet link: https://documentation.espressif.com/esp32_datasheet_en.pdf

Source code link: https://drive.google.com/drive/folders/1WCk-JCeRJgMnoQPy-eDcAzW4VPxeii7w?usp=drive_link

;